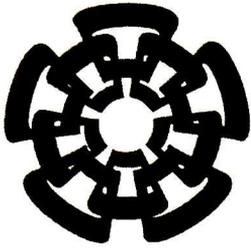


CT-687-SS1

Don - 2012

xx(199689.i)



Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional
Unidad Guadalajara

Desarrollo del procesador de banda base del estándar Bluetooth

Tesis que presenta:

Abisaí Ramírez Pérez
para obtener el grado de:

Maestro en Ciencias
en la especialidad de:

Ingeniería Eléctrica
Directores de Tesis

Dr. Ramón Parra Michel
Dr. Luis Fernando González Pérez



CLASIF.	CT00591
ADQUIS.	CT-687-SS1
FECHA:	28-08-2012
PROCED.	Dom-2012
	\$

ID: 199530-2001

Desarrollo del procesador de banda base del estándar Bluetooth

**Tesis de Maestría en Ciencias
Ingeniería Eléctrica**

Por:

Abisaí Ramírez Pérez

Ingeniero en Electrónica

Instituto Tecnológico de Veracruz 2002-2006

Becario de CONACYT, expediente no. 40550

Directores de Tesis

Dr. Ramón Parra Michel

Dr. Luis Fernando González Pérez

Resumen

Esta tesis presenta el desarrollo del estándar Bluetooth dentro del proyecto ASTECAS (por las siglas en inglés *A SoFTwarE defined radio receiver based on a Configurable DSP and An RF Sigma-Delta ADC*). La idea principal del proyecto es crear un ASIC versátil (por las siglas en inglés *Application-Specific Integrated Circuit*); en este ASIC se incluirán los estándares Bluetooth, ZigBee y WiFi en la misma pieza de silicio, y por software el usuario definirá el estándar a utilizar. Una ventaja significativa de esta implementación será la capacidad de reutilizar el hardware común, esto optimizará área. En este proyecto colaboran el CINVESTAV-IPN Unidad Guadalajara, Instituto Tecnológico y de Estudios Superiores de Monterrey, la Universidad Pierre & Marie Currie en París y la compañía Flexras Technologies.

Particularmente, esta tesis presenta un modelo de oro para el estándar Bluetooth V1.2 en el procesamiento de banda base y la capa física. En el procesamiento banda base incluye HEC, FEC, creación de paquete, código de acceso, palabra de sincronía, preámbulo, etc. Este modelo permitirá la capacidad de prueba del futuro hardware, el cual cumpliría la especificación Bluetooth V1.2. Para la capa física, este trabajo presenta un análisis de las implementaciones digitales de diferentes moduladores GFSK. Así mismo, se realizó un análisis de las implicaciones de la implementación digital de diferentes demoduladores y se verificó su desempeño. Encontrando que la mayoría de ellos fueron diseñados para implementaciones analógicas, por esta razón, se desarrolló un nuevo algoritmo de demodulación GFSK digital. Basado en estos análisis y simulaciones de desempeño se encontró el mejor par modulador – demodulador para GFSK, que pudiera cumplir con los requerimientos de implementación digital y desempeño.

Abstract

This thesis presents a Bluetooth standard model which is part of the ASTECAS project (A Software defined radio receiver based on a Configurable DSP and An RF Sigma-Delta ADC). The project main idea is to create a versatile ASIC (Application-Specific Integrated Circuit), in this ASIC will include Bluetooth, ZigBee and WiFi standards in the same silicon piece and by software the user will define the usable standard. A significant advantage of this implementation will be the capability to reuse common hardware, this will optimize area. In this project collaborates CINVESTAV- IPN Guadalajara unit, Institute of Technology and higher education of Monterrey, the University Pierre & Marie Curie at Paris and Flexras Technologies Company.

Mainly, this work presents a golden model for Bluetooth standard V1.2 in the baseband processing and physical layer. The baseband processing includes HEC, FEC, CRC, Packet creation, Access Code, Sync Word, Preamble, etc. This model will allow testability of future hardware which could fit the Bluetooth Spec V1.2. For the physical layer, this work presents a digital implementation analysis for different GFSK modulators. In the same way, a digital implementation analysis and a performance test were realized for GFSK demodulators. Finding that most of them were designed for analog implementation, for this reason, a new digital GFSK demodulator algorithm was developed. Based on these analysis and performance simulations, the best GFSK pair modulator- demodulator was found which could fit performance and digital implementation requirements.

Agradecimientos

A mis padres Don Francisco y Doña Irene por tener la dicha de tenerlos conmigo en cada paso que doy y por su apoyo incondicional, consejos, enseñanzas y amor. A mis hermanos Iligan, Horacio, Paco, Laura, Noé y Gustavo.

To my sweet heart Marisol por estar en cada momento acompañándome en este camino.

A mis asesores Dr. Ramón Parra y Dr. Luis Fernando González por sus enseñanzas, consejos, apoyo, confianza y por darme la oportunidad de poner mi granito de arena en el mejoramiento de nuestra sociedad al cambiarme la visión de lo que debemos hacer por nuestro México.

A todos mis profesores del CINVESTAV-IPN Guadalajara.

A mis compañeros de generación de Tele 09.

A CONACYT por el apoyo económico.

Contenido

Capítulo 1 Introducción a Bluetooth	1
1.1 Introducción.....	1
1.1.1 Justificación	2
1.2 Antecedentes	4
1.3 Principio de funcionamiento de la tecnología inalámbrica Bluetooth.....	5
1.4 Estado del arte	8
1.5 Definición del problema	9
1.6 Objetivos	9
1.6.1 Objetivo general.....	9
1.6.2 Objetivo específico	9
Capítulo 2 Especificación del sistema Bluetooth V1.2.....	11
2.1 Especificación de radiofrecuencia	11
2.1.1 Banda de frecuencias y arreglo de canales	11
2.1.2 Características del transmisor	11
2.1.2.1 Características de modulación	12
2.1.3 Características del receptor	12
2.2 Especificación banda base.....	13
2.2.1 Descripción general	13
2.2.1.1 Reloj Bluetooth.....	14
2.2.1.2 Direccionamiento del dispositivo Bluetooth.....	14
2.2.1.3 Códigos de acceso.....	15
2.2.2 Canal físico	16
2.2.2.1 Canal físico de piconet básico.....	17
2.2.2.2 Canal físico de piconet adaptado	18

2.2.2.3 Canal físico page scan.	18
2.2.2.4 Canal físico inquiry scan.....	19
2.2.2.5 Selección de salto.	20
2.2.2.5.1 Esquema de selección general.	20
2.2.2.5.2 Kernel de selección.....	23
2.2.2.5.3 Kernel de selección de salto adaptado.	26
2.2.2.5.4 Palabra de control.....	27
2.2.3 Enlace físico.....	28
2.2.3.1 Supervisión del enlace.	28
2.2.4 Transporte lógico.	29
2.2.4.1 Dirección de transporte lógico (LT_ADDR).	29
2.2.4.2 Rutinas de Transmisión / Recepción.	30
2.2.5 Enlaces lógicos.	31
2.2.6 Paquetes.....	32
2.2.6.1 Formato general.	32
2.2.6.2 Orden de bits.....	32
2.2.6.3 Código de acceso.....	32
2.2.6.3.1 Preámbulo.....	33
2.2.6.3.1 Palabra de sincronía.	34
2.2.6.3.2 Trailer.....	35
2.2.6.4 Encabezado de paquete.....	36
2.2.6.5 Tipos de paquetes.	37
2.2.6.6 Formato de payload.....	38
2.2.6.6.1 Campo de datos síncrono.	38
2.2.6.6.2 Campo de datos asíncrono.	38

2.2.7 Procesamiento del flujo de bits.	39
2.2.7.1 Verificación de errores	40
2.2.7.1.1 Generación de HEC.....	40
2.2.7.1.2 Generación de CRC	40
2.2.7.2 Whitening de datos.....	41
2.2.7.3 Corrección de errores.....	41
2.2.7.3.1 FEC 1 / 3.....	42
2.2.7.3.2 FEC 2 / 3.....	42
2.2.7.3.3 Esquema ARQ.....	42
2.2.8 Operaciones del controlador de enlace.	43
2.2.8.1 Estado STANDBY	43
2.2.8.2 Sub-estados de establecimiento de conexión.	43
2.2.8.2.1 Sub-estado de escaneo de page.	44
2.2.8.2.2 Sub-estado de page.	44
2.2.8.2.3 Sub-estado de respuesta de page.....	45
2.2.8.3 Sub-estados de descubrimiento de dispositivo.	46
2.2.8.3.1 Sub-estados de escaneo de inquiry.....	47
2.2.8.3.2 Sub-estados de Inquiry.....	47
2.2.8.3.3 Sub-estados de respuesta de inquiry.	47
2.2.8.4 Estado CONNECTION	48
2.2.8.5 Estado PARK.	48
Capítulo 3 Descripción del Modelo de Oro.....	49
3.1 Introducción.....	49
3.2 Arquitectura del procesador banda base.....	50
3.2.1 Máquina de control de enlace.....	51
3.2.2 Buffer o memoria de puerto dual.	52

3.2.3	Generador de palabra de sincronía.....	52
3.2.4	Control de timing & CLK offset.....	53
3.2.5	Máquina de selección de saltos.....	53
3.2.6	Sincronía de trama.....	54
3.2.7	Control de extracción de DC & ganancia.....	54
3.2.8	Ensamblador de paquetes.....	54
3.3	Modulación / demodulación de la señal de fase continua digital GMSK..	58
3.3.1	Características de la modulación GMSK.....	58
3.3.1.1	Representación matemática de la señal GMSK.....	60
3.3.1.2	Arquitectura del modulador GMSK.....	62
3.3.1.3	PSD de la señal GMSK generada por los moduladores.....	64
3.3.2	Demodulador GMSK.....	65
3.3.2.1	Arquitectura del receptor GMSK.....	69
3.3.2.2	MLBE.....	71
3.3.2.3	Desempeño de BER del demodulador GMSK.....	76
3.3.2.3.1	BER variando la cantidad N de símbolos observados.....	76
3.3.2.3.2	BER variando la longitud L de la respuesta del pulso $g(t)$... 77	77
3.3.2.3.3	BER variando el índice de modulación h	79
3.3.2.3.4	BER variado técnica de modulación.....	80
3.3.2.4	Arquitectura del MLBE para $N = 3$ y $N = 2$	82
Capítulo 4	Pruebas y resultados.....	85
4.1	Introducción.....	85
4.2	Pruebas al generador de tramas Bluetooth.....	85
4.2.1	Pruebas de caja blanca.....	85
4.2.2	Pruebas de caja negra.....	86

4.3 Análisis en punto fijo del demodulador GMSK.	90
4.3.1 Curvas de BER del demodulador GMSK en punto fijo.....	94
4.4 Pruebas de interconexión del procesador banda base.	94
Capítulo 5 Conclusiones y Trabajo Futuro	99
5.1 Conclusiones.....	99
5.2 Trabajo futuro.....	100
GLOSARIO.....	101
ACRÓNIMOS	102
BIBLIOGRAFÍA	104

Índice de figuras

Figura 1.1 Arquitectura general del system core de Bluetooth V1.2 [4].	7
Figura 2.1 a) Piconets con operación de un solo esclavo, b) operación con múltiples esclavos y c) operación scatternet [4].	13
Figura 2.2 Reloj Bluetooth [4].	14
Figura 2.3 Formato de la dirección de dispositivo Bluetooth (BD_ADDR) [4].	15
Figura 2.4 Obtención de CLK en el maestro (a) y en el esclavo (b) [4].	18
Figura 2.5 Obtención del reloj CLKE para paging [4].	19
Figura 2.6 Diagrama a bloque del esquema de selección de salto [4].	21
Figura 2.7 Esquema de selección de salto en el estado CONNECTION [4].	22
Figura 2.8 Ejemplo del mecanismo del mismo canal [4].	23
Figura 2.9 Diagrama a bloques del kernel de selección de saltos [4].	23
Figura 2.10 Operación XOR para el sistema de saltos [4].	24
Figura 2.11 Operación Butterfly para el caso de $P0$.	25
Figura 2.12 Operación de permutación para el sistema de salto [4].	25
Figura 2.13 Diagrama a bloques del mecanismo de selección de salto adaptativo [4].	26
Figura 2.14 Control para el sistema de saltos [4].	27
Figura 2.15 Diagrama funcional del buffer de transmisión.	30
Figura 2.16 Formato del paquete general	32
Figura 2.17 Formato del código de acceso.	33
Figura 2.18 Contrucción de la palabra de sincronía [4].	34
Figura 2.19 El LFSR y el estado inicial para generar la secuencia PRNG [4].	35
Figura 2.20 Trailer en CAC: a) MSB de la palabra de sincronía es 0, y b) MSB es 1.	35
Figura 2.21 Formato del encabezado [4].	36
Figura 2.22 Encabezado de payload para paquetes ACL de una ranura [4].	38
Figura 2.23 Encabezado de payload para paquetes ACL de multi-ranura [4].	38
Figura 2.24 Procesamiento del encabezado	39
Figura 2.25 Procesamiento del payload.	39

Figura 2.26 Circuito LFSR para la generación del HEC [4].	40
Figura 2.27 Circuito LFSR para la generación de CRC [4].	41
Figura 2.28 Circuito de LFSR del whitening de datos [4].	41
Figura 2.29 Generador LFSR para el código Hamming (15, 10) [4].	42
Figura 2.30 Diagrama de estados del controlador de enlace [4].	43
Figura 3.1 Top level del procesador banda base.	50
Figura 3.2 Bloques internos del procesador banda base.	51
Figura 3.3 Diagrama a bloques del ensamblador de paquetes.	55
Figura 3.4 Espectro de densidad de potencia de GMKS [11].	58
Figura 3.5 Desempeño de BER en un canal AWGN [11].	59
Figura 3.6 Implementación del Generador de la señal GMSK con ROM.	63
Figura 3.7 Densidad espectral de potencia normalizada para 3 moduladores GMSK.	65
Figura 3.8 Desempeño BER vs EbNo del demodulador de cruce por ceros I&D para diferentes moduladores.	67
Figura 3.9 Desempeño BER vs EbNo de los demoduladores Delay&Multiply y Quadricorrelator en banda base contra diferentes moduladores.	68
Figura 3.10 Demodulación coherente de la señal GMSK recibida.	69
Figura 3.11 Arquitectura de demodulador GMSK en banda base.	70
Figura 3.12 Desempeño del demodulador variando el numero de símbolos observados N .	76
Figura 3.13 Desempeño del demodulador GMSK variando la longitud L del pulso Gaussiano y $N = 3$.	77
Figura 3.14 Desempeño del demodulador GMSK variando L del pulso Gaussiano y $N = 2$.	78
Figura 3.15 BER vs EbNo del demodulador GMSK, variando h y $N = 3$.	79
Figura 3.16 BER vs EbNo del demodulador GMSK, variando el índice de modulación h y $N = 2$.	80
Figura 3.17 Curva de BER vs EbNo del desempeño del demodulador con modulación GFSK.	81

Figura 3.18 Desempeño del demodulador con modulación GMSK por medio de VCO.	81
Figura 3.19 Arquitectura del bloque de estimación de máxima semejanza (MLBE) con $N = 3$	83
Figura 3.20 Arquitectura del bloque de estimación de máxima semejanza (MLBE) con $N = 2$	84
Figura 3.21 Aproximación de los multiplicadores por sumas y corrimientos.....	84
Figura 4.1 Uso de la función "fi" en MATLAB.....	90
Figura 4.2 Bloques generales del demodulador GMSK.....	91
Figura 4.3 BER vs EbNo comparativo para determinar la cantidad de bits fraccionarios.....	92
Figura 4.4 Bloques del demodulador GMSK en punto fijo.....	93
Figura 4.5 Desempeño del demodulador GMSK en punto fijo.....	94
Figura 4.6 Diagrama de la interconexión de los bloques para las pruebas.....	95
Figura 4.7 Formas de ondas de la señal GMSK en fase y en cuadratura para un paquete HV1.....	96
Figura 4.8 Formas de onda de la señal GMSK, para un paquete HV1 con ruido.....	96

CAPÍTULO 1 INTRODUCCIÓN A BLUETOOTH

1.1 Introducción

El constante crecimiento de las tecnologías móviles ha llevado a la tecnología Bluetooth a tener esta misma tendencia de crecimiento en sus más de 12 años de existencia desde que fue adoptado por el Bluetooth SIG (por las siglas en inglés *Special Interest Group*), al pasar de 1 millón de unidades de Bluetooth puestas en uso por semana en 2000, a más de 30 millones de unidades por semana en 2010. Por lo que actualmente existen cerca de 3 billones de dispositivos Bluetooth en el mercado [1].

Esto ha llevado a tener dispositivos Bluetooth dentro de una inmensidad de aparatos electrónicos, tales como dispositivos audiovisuales por ejemplo cámaras o videocámaras, diademas inalámbricas, teléfonos, PDA's, accesorios de teléfonos móviles, se ha inmerso en el ramo automotriz, dispositivos de entrada como ratones y teclados, consolas de videojuegos, dispositivos médicos y computadoras personales.

Bluetooth es un estándar de comunicaciones inalámbricas de corto alcance, cuya intención inicial era reemplazar los cables de dispositivos electrónicos fijos o portables utilizados en el intercambio de datos. Actualmente Bluetooth es más que el simple reemplazo de cables, ya que permite la comunicación en piconet o scatternet, es decir, redes de Bluetooth sin la necesidad de una estación base, access point o cualquier otro dispositivo de conmutación.

Ante el impresionante crecimiento de mercado que ha tenido la tecnología Bluetooth y las múltiples aplicaciones que se le ha encontrado y se le encontrarán, nos planteamos la pregunta ¿Cuales han sido los avances obtenidos desde que se implementó esta tecnología hace más de 12 años en el desarrollo e investigación de la tecnología Bluetooth en México?

1.1.1 Justificación

El atraso científico y tecnológico en el que se encuentra inmerso nuestro país ha generado una brecha abismal, entre los países que le han apostado fuertemente al desarrollo de la ciencia y tecnología y el nuestro. Esto principalmente se debe a la falta de un plan nacional para el desarrollo de las mismas, lo que nos ha relegado al papel de consumidores de tecnología.

Las tecnologías que llevan una tendencia de rápido crecimiento son los sistemas de comunicaciones inalámbricos, entre estos sistemas podemos nombrar: 60 GHz donde se consideran altas tasas de transferencia de datos; WiMax, usado para conexiones de Internet metropolitano; WiFi, ha mostrado su utilidad en las conexiones de Internet de casa; ZigBee, usado en redes inalámbricas de área personal que requieren conexiones seguras a baja tasa de envío de datos; Bluetooth, una tecnología que tiene más de 12 años en el mercado y que no ha sido desarrollado en México, por lo que se carece de un modelado del sistema y por ende de las implementaciones en hardware.

Por esta razón, un integrado multi-estándar que cuente con algunos de estos sistemas de comunicaciones sería una ventaja competitiva en el mercado de las tecnologías. Un proyecto que pretende atacar esta temática se denomina ASTECAS (por las siglas en inglés *A SofIwarE defined radio receiver based on a Configurable DSP and An RF Sigma-Delta ADC*), en él se pretenden incluir los estándares WiFi, ZigBee y Bluetooth. En este proyecto colaboran, la Universidad Pierre & Marie-Curie (UPMC), la empresa Flexras Technologies, Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) y el Centro de Investigaciones y de Estudios Avanzados del I.P.N. (CINVESTAV) Unidad Guadalajara.

Con base en este proyecto, el modelado del procesador banda base del estándar Bluetooth se hace imperativo, así como el análisis para determinar el mejor par modulador demodulador usado en la transmisión y recepción de datos. El procesador banda base, en conjunto con la sección de radiofrecuencia desarrollada en la UPMC, constituirán el controlador Bluetooth. Los bloques comunes al

controlador Bluetooth y a los otros dos estándares, se reutilizarán bajo una aproximación de diseño SoC (por las siglas en inglés *System On Chip*).

1.2 Antecedentes

En sus inicios la tecnología Bluetooth fue creada como reemplazo para el protocolo RS-232 tanto para los dispositivos fijos como para los portables y fue creada por Ericsson en 1994 [2] para el año de 1998 se formó un grupo para la promoción y publicación del estándar Bluetooth denominado Bluetooth SIG, este grupo inicialmente fue conformado por 5 compañías, las cuales son: Ericsson, IBM, Intel, Nokia y Toshiba. Actualmente este grupo está conformado por más de 13 000 compañías miembro [3] entre los que se encuentran líderes en telecomunicaciones, informática, consumidores electrónicos, automotriz, automatización industrial e industrias de redes.

El nombre Bluetooth proviene de un rey danés del siglo X, Harald Bluetooth o Haral I de Dinamarca y parte de Noruega, que unificó varias tribus danesas en un sólo reino, donde las guerras y los clanes feudales prevalecían.

Los fundadores del grupo Bluetooth SIG, adoptaron este nombre dado que se adaptaba debido a que la tecnología Bluetooth fue desarrollada en Escandinavia y, por que posibilita unir diferentes industrias como la telefonía celular, informática, y el mercado de automóviles, aunque actualmente los usos del Bluetooth se han masificado.

El logo une los caracteres "H" parecido a un asterisco, y la "B" del alfabeto rúnico que son las iniciales de Haral Blatand ó Haral Bluetooth en inglés.

A un año de la conformación del Bluetooth SIG se publicó la especificación técnica Bluetooth V1.0, y no fue hasta el año de 2004 cuando fue introducida la especificación Bluetooth Versión 2.0 + EDR de las siglas en inglés *Enhanced Data Rate*. A finales del siguiente año fue un hit del mercado, ya que mejora las velocidades de transferencia de datos. La especificación del Bluetooth Versión 2.1 + EDR es adoptada por el Bluetooth SIG en 2007. para el año de 2009 se introduce la especificación del Bluetooth Versión 3.0 + HS de las siglas en inglés High Speed.

La adopción de la especificación Bluetooth 4.0 no fue hasta el año 2010, la cual incluye una prominente característica de "tecnología de baja energía".

1.3 Principio de funcionamiento de la tecnología inalámbrica Bluetooth

Para los alcances de la presente tesis se tomará como referencia la especificación del estándar Bluetooth Versión 1.2.

El sistema de Bluetooth consiste de un transceptor de radio frecuencia, la banda base y la pila de protocolos. Este sistema ofrece servicios que posibilitan la conexión de dispositivos y el intercambio de una variedad de tipos de datos entre estos dispositivos.

La parte de radio frecuencia de Bluetooth (capa física) opera en la banda ISM (por las siglas en inglés *Industrial, Scientist Medical*) con ancho de banda de 2400-2483.5 MHz. Este sistema utiliza un transceptor con saltos de frecuencia para combatir interferencia y desvanecimiento, este método de conmutación de portadora entre varios canales de frecuencia se denomina FHSS (por las siglas en inglés *Frequency Hopping Spread Spectrum*). En la sección de RF (por las siglas en inglés *Radiofrequency*) se utiliza el esquema de modulación GFSK (por las siglas en inglés *Gaussian Frequency Shift Keying*), donde la tasa de bits resulta ser de 1 Megabits por segundo (Mbps).

Durante la operación normal un canal de radio físico es compartido por un grupo de dispositivos que son sincronizados a un reloj común y a un patrón de saltos de frecuencia. El dispositivo que proporciona la referencia es el maestro, y los demás dispositivos sincronizados al reloj del maestro y al patrón de saltos de frecuencia son los esclavos. A este grupo de dispositivos sincronizados se denomina piconet.

El patrón de saltos de frecuencia es determinado algorítmicamente por ciertos campos de la dirección del Bluetooth y el reloj del maestro, el patrón de saltos básico es un ordenamiento pseudo-aleatorio de las 79 frecuencias de la banda ISM, empezando en 2402 MHz hasta 2480 MHz. Este patrón de saltos puede ser adaptado para excluir una porción de las frecuencias que son usadas por dispositivos interferentes, esto permite mejorar la coexistencia con sistemas que utilizan la misma banda ISM.

El canal físico es subdividido en unidades de tiempo llamadas slots. Por lo que los datos son transmitidos entre dispositivos Bluetooth en paquetes, que son posicionados en estos slots. Cuando las circunstancias lo permiten, un solo paquete puede tener un número consecutivo de slots. Los saltos de frecuencia se dan entre la transmisión o recepción de paquetes. Un efecto de una transmisión full dúplex es proveído por el dispositivo Bluetooth por medio del uso del esquema TDD (por las siglas en inglés *Time-Division Duplex*).

La jerarquía de canales y enlaces desde el canal físico hacia arriba son canal físico, enlace físico, transporte lógico, enlace lógico y canal L2CAP (por las siglas en inglés *Logical Link Control and Adaptation Protocol*).

Un enlace físico es formado entre dos dispositivos que transmiten paquetes en cualquier dirección entre ellos. En una piconet, existe un enlace físico entre cada esclavo y el maestro, estos no son formados entre esclavos.

El enlace físico es usado como un transporte para uno o más enlaces lógicos que soportan tráfico síncrono "unicast", asíncrono e isócrono y tráfico de "broadcast" El tráfico sobre enlaces lógicos es conmutado dentro del enlace físico por medio de la utilización de slots asignados por una función de calendarización en el administrador de recursos.

Un protocolo de control para las capas de banda base y física es transportado por medio de enlace lógico en adición a los datos de usuario. Este es el LMP de las siglas en inglés *Link Manager Protocol*. Los dispositivos que están activos en una piconet tienen un transporte lógico asíncrono orientado a conexión (ACL de las siglas en inglés *asynchronous connection-oriented*) que es usado para transportar la señalización del protocolo LMP. Este es conocido como transporte lógico ACL. El transporte lógico ACL por omisión es creado cuando el dispositivo se adjunta a una piconet. Transporte lógico adicional puede ser creado para transportar flujos de datos síncronos cuando se requiere.

La función del Administrador del Enlace de las siglas en inglés Link Manager usa el LMP para controlar las operaciones de dispositivos en la piconet y proveer servicios para administrar las capas inferiores (capa de radio y capa de banda base).

EL protocolo LMP sólo es transportado por el transporte lógico ACL por omisión y el transporte lógico de difusión por omisión.

Arriba de la capa de banda base, la capa L2CAP provee un canal para aplicaciones y servicios. Esto conlleva la segmentación y reensamble de datos de aplicación, así como el multiplexaje y demultiplexaje de múltiples canales sobre un enlace lógico compartido. El L2CAP tiene un canal de control de protocolo que es transportado sobre el transporte lógico ACL por omisión.

Para tener una imagen más concreta de las interacciones entre las capas y su estructura se muestra la Figura 1.1.

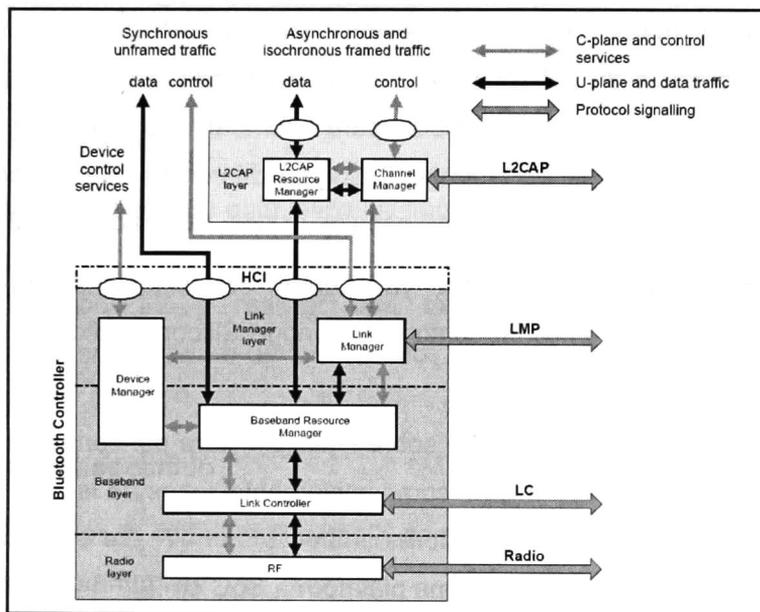


Figura 1.1 Arquitectura general del system core de Bluetooth V1.2 [4].

1.4 Estado del arte

La última versión de la especificación del estándar Bluetooth fue adoptada en 2010 por el Bluetooth SIG y se denomina Bluetooth Core Specification Version 4.0, y cuenta con la nueva característica de tecnología de baja energía. Esta nueva especificación incluye las características de BR/EDR de las siglas en inglés *Basic Rate/ Enhance Data Rate* la cual en el modo básico permite tasas de datos de 721.2 Kbps y de 2.1 Mbps para el modo EDR. La nueva característica se denomina LE (por las siglas en inglés *Low Energy*), esta cualidad está dirigida a los dispositivos que requieren un bajo consumo de corriente, baja complejidad y bajo costo.

De la misma manera que en la versión básica, el módulo de radiofrecuencia opera en la frecuencia de 2.4 GHz conocida como banda ISM, se utiliza el mismo transceptor de saltos de frecuencia para combatir la interferencia y los desvanecimientos, y provee portadoras FHSS. La diferencia con la versión básica radica en que en esta versión LE son 40 canales físicos, separados por 2 MHz.

Actualmente existen grupos que ya cuentan con soluciones para la nueva especificación de la versión 4, tal como Chambridge Silicon Radio con su producto CSR μ Energy CSR1000/1 [5].

Se han encontrado aportaciones sobre la arquitectura del controlador Bluetooth para la versión 2.1 en [6]. Así mismo como implementaciones de las capas inferiores del controlador Bluetooth en procesadores embebidos en [7]. Se encontró reportado implementaciones de Bluetooth bajo una plataforma SoC en [8].

Como todo desarrollo tecnológico, el sistema Bluetooth se debe codificar en lenguaje de alto nivel para describir su comportamiento, así mismo, se debe realizar la implementación en lenguaje de descripción de hardware. El aporte principal de la descripción del comportamiento es conocer los bloques que conforman el sistema y plantear la reutilización de bloques comunes a otros estándares, con la idea de crear un chip multi-estándar bajo el concepto de SoC.

1.5 Definición del problema.

El nulo avance nacional de la tecnología inalámbrica Bluetooth y la necesidad de contar con una implementación de procesamiento de banda base del estándar Bluetooth. El procesador banda base en conjunto con el módulo de radiofrecuencia, el administrador de enlace y de manera opcional el HCI (por las siglas en inglés *Host Controller Interface*) conformarán el controlador de Bluetooth. El controlador Bluetooth podrá ser integrado dentro del un mismo ASIC en combinación con sus similares WiFi y ZigBee, para ello se hace imprescindible el desarrollo de un modelo de referencia para su futura implementación y una exploración de las posibles arquitecturas de implementación en hardware.

1.6 Objetivos

1.6.1 Objetivo general

El objetivo general de esta investigación es crear un modelo en punto fijo del transmisor y receptor del módulo de procesamiento de banda base del estándar Bluetooth Versión 1.2 y realizar una exploración de las arquitecturas de implementación en hardware de los módulos que constituyen el procesador banda base.

1.6.2 Objetivo específico

1. Estudiar el estándar Bluetooth para conocer los requerimientos para su implementación.
2. Realizar un modelo de oro en punto fijo del transmisor y receptor del módulo de banda base del estándar Bluetooth, en lenguaje de cómputo técnico de alto nivel MATLAB
3. Validar los datos obtenidos de las simulaciones del procesador banda base con los datos muestra proporcionados con el estándar.
4. Hacer un planteamiento de la arquitectura del procesador banda base.
5. Explorar las arquitecturas para el modulador GMSK y ver la viabilidad de su implementación en hardware digital.

6. Explorar las arquitecturas para el demodulador GMSK para observar la viabilidad de su implementación en hardware digital, y que cumpla con los requerimientos de BER definido en el estándar.
7. Proponer un par modulador – demodulador GFSK, que cumple con los requerimientos del estándar Bluetooth V1.2, para su implementación en hardware.

CAPÍTULO 2 ESPECIFICACIÓN DEL SISTEMA BLUETOOTH V1.2

2.1 Especificación de radiofrecuencia.

Los dispositivos Bluetooth operan en la banda no licenciada ISM de 2.4 GHz, utilizando un transceptor con saltos de frecuencia para combatir la interferencia y los desvanecimientos. Una modulación GFSK es aplicada para reducir la complejidad del transceptor, a una tasa de símbolos de 1 Mega símbolo/seg. El cometido de este capítulo es definir los requerimientos para el radio de Bluetooth.

2.1.1 Banda de frecuencias y arreglo de canales.

De la banda ISM ubicada en el rango de 2400 – 2483.5 MHz, se usa un ancho de banda de 79 MHz para realizar los saltos de frecuencia, en canales de 1 MHz de ancho de banda.

Rango regulado	Canales de RF
2.400 - 2.4835 GHz	$f = 2402 + k$ MHz, $k = 0, \dots, 78$

Tabla 2.1 Banda de frecuencias y canales de RF.

Los canales de RF están espaciados 1 MHz y se encuentran numerados en canales k, como se muestra en la Tabla 2.1. Para cumplir con las regulaciones fuera de banda en cada país, se usan bandas de guarda en los bordes inferior y superior:

Banda de guarda inferior	Banda de guarda superior
2 MHz	3.5 MHz

Tabla 2.2. Bandas de guarda.

2.1.2 Características del transmisor.

Los dispositivos Bluetooth son clasificados dentro de 3 clases de acuerdo con su potencia de salida:

Clase de potencia	Potencia de salida máxima	Potencia de salida nominal	Potencia de salida mínima	Control de potencia
1	100 mW (20 dBm)	N/A	1 mW (0 dBm)	Pmin<+4 dBm a Pmax Opcional: Pmin a Pmax
2	2.5 mW (4 dBm)	1 mW (0 dBm)	0.25 mW (-6 dBm)	Opcional: Pmin a Pmax
3	1 mW (0 dBm)	N/A	N/A	Opcional: Pmin a Pmax

Tabla 2.3 Clases de dispositivos de acuerdo a su potencia de salida.

Pmin = Limite de potencia inferior Pmin < -30 dBm es sugerido pero no mandatorio.

Los dispositivos clase 1 deben implementar control de potencia. El control de potencia debe ser implementado para limitar la potencia transmitida con dispositivos con más de +4 dBm. Con dispositivos bajo este límite es opcional y puede ser usado para optimizar el consumo de potencia.

2.1.2.1 Características de modulación.

El esquema de modulación utilizado es GMSK con un producto de periodo de bit por ancho de banda $BT = 0.5$. El índice de modulación debe estar entre 0.28 y 0.35. El valor uno binario se representa por una desviación positiva de frecuencia, y un cero binario es representado por medio de una desviación negativa de frecuencia.

2.1.3 Características del receptor.

El nivel de sensibilidad actual es definido como el nivel de entrada para el cual se logra un BER (por las siglas en inglés *Bit Error Rate*) de 0.1%. La sensibilidad del receptor debe ser menor o igual a -70 dBm con cualquier transmisor Bluetooth compatible con las especificaciones, esto se logra teniendo un valor de E_b/N_0 de 21 dB de acuerdo a [9].

2.2 Especificación banda base.

2.2.1 Descripción general.

Esta sección realiza una descripción del procesamiento en banda base del estándar Bluetooth V 1.2.

El sistema Bluetooth tiene la capacidad de ofrecer conexiones punto a punto ó punto a multipunto. En el caso de las conexiones punto a punto, el canal físico es compartido entre 2 dispositivos Bluetooth; en las conexiones punto a multipunto, el canal físico es compartido entre varios dispositivos Bluetooth. Dos o más dispositivos compartiendo el mismo canal físico forman una *piconet*. Un dispositivo Bluetooth actúa como maestro de la piconet, donde los demás dispositivos actúan como esclavos. Hasta 7 esclavos pueden estar activos en una piconet, y muchos más esclavos pueden permanecer conectados en un estado de estacionado (parked), estos dispositivos permanecen solo sincronizados al maestro y pueden convertirse en dispositivos activos. El acceso al canal es controlado por el maestro

Las piconets que tienen dispositivos en común son llamadas "*scatternet*". Cada piconet tiene su maestro, sin embargo, los esclavos pueden participar en diferentes piconets por conmutación por división de tiempo. Además, un maestro en una piconet puede ser esclavo en otra piconet. Las frecuencias de las piconets no están sincronizadas, y cada piconet utiliza su propia secuencia de salto.

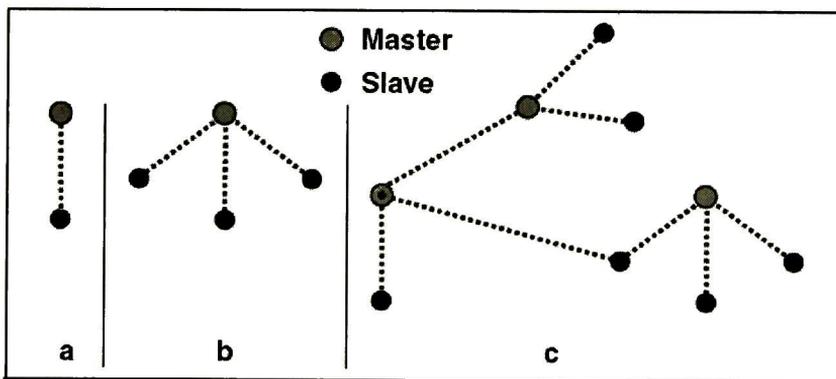


Figura 2.1 a) Piconets con operación de un solo esclavo, b) operación con múltiples esclavos y c) operación scatternet [4].

2.2.1.1 Reloj Bluetooth

Cada dispositivo Bluetooth tiene un reloj nativo que debe ser derivado de un reloj de sistema de corrida libre. Para realizar la sincronización con otros dispositivos Bluetooth, se utilizan compensaciones las cuales son sumadas al reloj nativo, y permite temporalmente que los relojes Bluetooth estén mutuamente sincronizados. Si el reloj es implementado con un contador, se requiere un contador de 28 bits. El bit menos significativo LSB (por las siglas en inglés Least Significant Bit) se incrementa cada $312.5 \mu\text{s}$ (es decir, la mitad de una ranura de tiempo). La frecuencia del reloj es de 3.2 KHz.

Cuatro periodos son importantes en el sistema Bluetooth: $312.5 \mu\text{s}$, $625 \mu\text{s}$, 1.25 ms y 1.28 s ; estos periodos corresponden a los bits del timer CLK0, CLK1, CLK2, CLK12, como se muestra en la Figura 2.2.

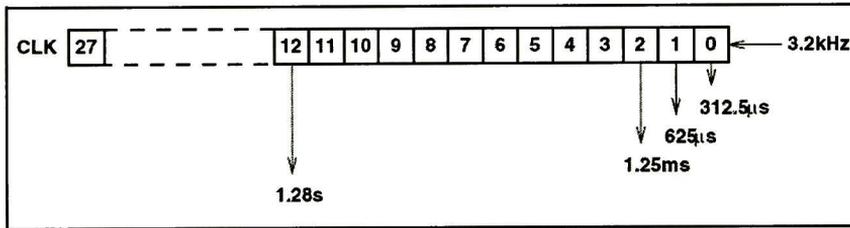


Figura 2.2 Reloj Bluetooth [4].

Dependiendo del modo y el estado en que se encuentre el dispositivo, el reloj tiene diferentes nombres, pero siempre teniendo como referencia el reloj nativo:

- CLKN reloj nativo
- CLKE reloj estimado
- CLK reloj maestro

Es importante hacer notar que el maestro nunca ajusta su reloj durante la existencia de la piconet.

2.2.1.2 Direccionamiento del dispositivo Bluetooth.

Cada dispositivo Bluetooth debe tener guardada una dirección de dispositivo Bluetooth única de 48 bits (BD_ADDR). Esta dirección es dividida en 3 campos:

- LAP: de las siglas en inglés *Low Address Part* el cual consiste de 24 bits.
- UAP: de las siglas en inglés *Upper Address Part* el cual consiste de 8 bits.
- NAP: de las siglas en inglés *Non-significant Address Part* el cual consiste de 16 bits.

En la Figura 2.3 se muestra un ejemplo de la división de los campos de la dirección del dispositivo Bluetooth.

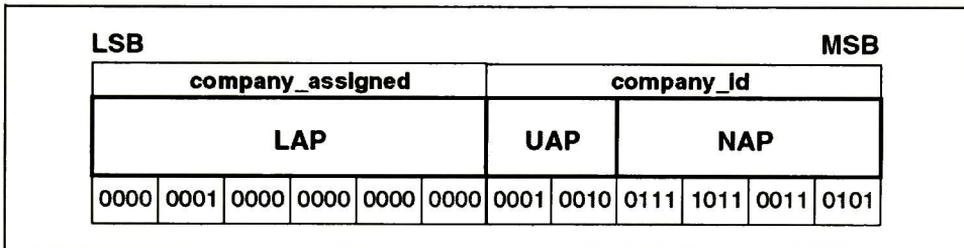


Figura 2.3 Formato de la dirección de dispositivo Bluetooth (BD_ADDR) [4].

La BD_ADDR puede tomar cualquier valor, excepto los 64 valores de LAP reservados para los estados “general inquiry” y “dedicated inquiry”

Un bloque de 64 LAP’s continuas es reservado para operaciones de “inquiry”; una LAP común para todos los dispositivos es reservada para *general inquiry*, las restantes 63 LAP’s son reservadas para *dedicated inquiry*, por esta razón, ninguna de estas LAP’s puede ser parte de la BD_ADDR del usuario.

Las LAP’s reservadas son en valor hexadecimal 0x9E8B00-0x9E8B3F La LAP de *general inquiry* es 0x9e8b33. Todas las direcciones en notación hexadecimal tienen el LSB en la posición derecha. Cuando una dirección LAP reservada es usada el campo de UAP es usado como DCI de las siglas en inglés *default check initialization*, en otro caso, es DCI es definido como 0x00.

2.2.1.3 Códigos de acceso.

En el sistema Bluetooth todas las transmisiones sobre el canal físico inician con un código de acceso. Se han definido 3 diferentes códigos de acceso para ello se tomaran los nombres del estándar en inglés:

Device Access Code (DAC)

Channel Access Code (CAC)

Inquiry Access Code (IAC)

El **device access code (DAC)** es usado durante los sub-estados **page**, **page scan**, y **page response**. El **channel access code (CAC)** es usado en el estado de **connection** y forma el inicio de todos los paquetes intercambiados sobre el canal físico de la piconet, este es obtenido de la LAP del BD_ADDR del maestro. El **inquiry access code** debe ser usado en el sub-estado de **inquiry**.

El código de acceso indica al receptor el arribo de un paquete. Es utilizado para la sincronización del temporizado y compensación. El receptor correlaciona contra la palabra de sincronización en el código de acceso, proporcionando robusta señalización.

2.2.2 Canal físico.

La capa inferior de la arquitectura del sistema Bluetooth es el **canal físico**. Todo canal físico se caracteriza por la combinación de una secuencia de saltos de frecuencia pseudo-aleatoria, el temporizado de ranuras de las transmisiones, el código de acceso y el codificado del paquete de cabecera. Para los canales físicos de piconet básicos y adaptados se usa cambios de frecuencia periódicos para reducir los efectos de interferencia y satisfacer los requerimientos regulatorios.

Para lograr la comunicación entre dos dispositivos, sus transceptores deben estar sintonizados a la misma frecuencia de RF al mismo tiempo, y estar dentro de un rango nominal uno de otro.

Cuatro canales físicos Bluetooth son definidos, cada uno optimizado y usado para un propósito diferente. Dos de estos canales físicos (canal de piconet básico y canal de piconet adaptado) son usados para la comunicación entre dispositivos conectados y están asociados a una piconet específica. Los restantes canales físicos son usados para el descubrimiento (**inquiry scan channel**) y conexión (**page scan channel**) de los dispositivos Bluetooth

Para poder soportar múltiples operaciones concurrentes, el dispositivo usa conmutación por división de tiempo entre los canales. De esta manera un dispositivo Bluetooth puede aparecer que opera en algunas piconets de manera simultánea.

Como se comentó en un párrafo superior, el canal físico lo define una secuencia de saltos de canales de RF pseudo-aleatoria, el temporizado del paquete y un código de acceso. La secuencia de salto es determinada por la UAP y LAP de la dirección del dispositivo Bluetooth y la secuencia de salto seleccionada. La fase en la secuencia de salto es determinada por el reloj Bluetooth. Todos los canales físicos son divididos en ranuras de tiempo cuya longitud es diferente dependiendo del canal físico. Por lo tanto cada transmisión o recepción es asociada con una ranura de tiempo. La máxima tasa de saltos es de 1600 saltos/seg en el estado **CONNECTION** y la máxima en los sub-estados de **inquiry** y **page** es de 3200 saltos/seg.

Los canales físicos se pueden clasificar de la siguiente manera:

- Canal físico de piconet básico.
- Canal físico de piconet adaptado.
- Canal físico page scan.
- Canal físico inquiry scan.

2.2.2.1 Canal físico de piconet básico.

Durante el estado de **CONNECTION** el canal físico de piconet básico es usado por defecto. Este mismo es definido por el maestro de la piconet, donde el maestro controla el tráfico sobre el canal físico de piconet por un esquema de *polling*.

Este canal físico se caracteriza por tener saltos pseudo-aleatorios dentro de los 79 canales de RF. Los saltos de frecuencia en el canal físico de piconet son determinados por el reloj de Bluetooth y la BD_ADDR del maestro. Cuando la piconet es establecida, el reloj del maestro es comunicado a los esclavos, dado que los relojes de los dispositivos son independientes, las compensaciones deben actualizarse de manera regular. La manera en que se realiza la compensación en el maestro y en el esclavo se muestra en la Figura 2.4. Todos los dispositivos en la piconet están sincronizados en tiempo y en los saltos de frecuencia.

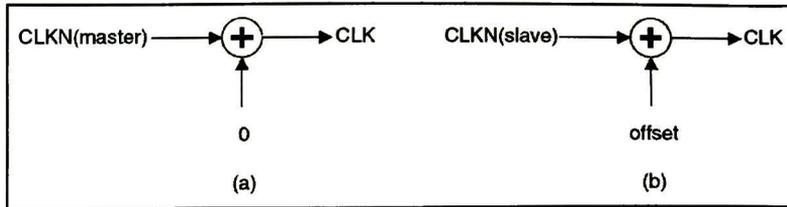


Figura 2.4 Obtención de CLK en el maestro (a) y en el esclavo (b) [4].

El canal físico de piconet básico es dividido en ranuras de tiempo de 625 μ s. Estas ranuras son numeradas de acuerdo a los 27 bits más significativos del reloj maestro Bluetooth (CLK28-1)

Un esquema TDD es usado cuando el maestro y el esclavo transmiten de manera alternada. Cada paquete inicia alineado con el inicio de la ranura de tiempo, la longitud máxima de los paquetes es de 5 ranuras de tiempo.

2.2.2.2 Canal físico de piconet adaptado.

El canal físico de piconet adaptado usa al menos N_{min} canales de RF (donde N_{min} es 20). Este canal puede ser usado para dispositivos conectados que tengan habilitado los saltos de frecuencia adaptativos AFH (por las siglas en inglés *adaptive frequency hopping*). Las principales diferencias con el canal básico es que la frecuencia de esclavo son las mismas que la frecuencia de la transmisión previa del maestro, y la otra diferencia es que el conjunto de frecuencias es menor a las 79 frecuencias del canal básico.

2.2.2.3 Canal físico page scan.

Aunque los roles de maestro y esclavo no están definidos antes de tener una conexión, el término maestro es usado para el dispositivo de paging (que se convierte en maestro en el estado **CONNECTION**) y esclavo es usado para el dispositivo de escaneo de page.

Un dispositivo de paging usa un estimado del reloj nativo del dispositivo de page scanning, CLKE; es decir, un offset debe ser sumado al CLKN del pager para aproximar el CLKN del receptor. CLKE deber ser derivado del CLKN de referencia sumando un offset. Usando el CLKN del receptor, el pager podría acelerar el

establecimiento de la conexión. La manera de obtener el CLKE se muestra en la Figura 2.5.

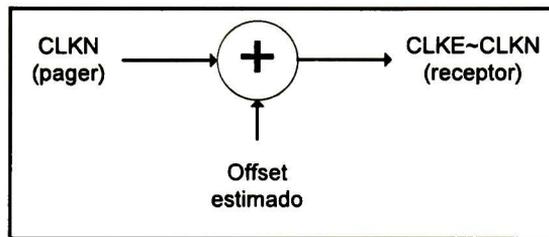


Figura 2.5 Obtención del reloj CLKE para paging [4].

El canal físico de page scan sigue un patrón de saltos más lento que el canal físico básico. El temporizado del canal page scan es determinado por el reloj Bluetooth nativo del dispositivo escaneado, así mismo la secuencia de saltos de frecuencia es determinado por la dirección Bluetooth del dispositivo escaneado.

Las secuencias de salto en el canal físico de page scan son: de page, de respuesta del maestro de page, de respuesta del esclavo de page, y de page scan. Debido a que el mensaje de paging es muy corto, la tasa de saltos es de 3200 saltos/seg.

2.2.2.4 Canal físico inquiry scan.

El término maestro es usado para el dispositivo de inquiring y el de esclavo para el dispositivo de escaneo de inquiry.

El reloj usado para inquiry y escaneo de inquiry deben ser los nativos.

En canal de escaneo de inquiry sigue un patrón de saltos más lento que el canal físico de piconet y es una secuencia pseudo-aleatoria corta a través de los canales de RF. El temporizado del canal es determinado por el reloj nativo Bluetooth del dispositivo escaneado mientras la secuencia de saltos de frecuencia es determinado por el código de acceso inquiry general.

Las secuencias de saltos utilizadas en este canal son: de inquiry, de respuesta de inquiry, y de escaneo de inquiry.

2.2.2.5 Selección de salto.

Son definidas seis tipos de secuencias de saltos, las cuales son:

- **Secuencia de salto page**, con 32 frecuencias de wake-up distribuidas igualmente sobre las 79 MHz, con un periodo de 32;
- **Secuencia de salto de respuesta de page** cubre 32 frecuencias que corresponden uno a uno con las secuencias de salto de page. El maestro y el esclavo usan diferentes reglas para obtener la misma secuencia.
- **Secuencia de salto de inquiry** con 32 frecuencias de wake-up distribuidas igualmente sobre los 79 MHz, con un periodo de 32.
- **Secuencia de salto de respuesta de inquiry** cubre 32 frecuencias que corresponden uno a uno con las secuencias de salto de inquiry.
- **Secuencia de salto de canal básico** tiene un periodo largo, no muestra patrones repetidos en periodos de tiempo corto, y distribuye los saltos de frecuencia sobre los 79 MHz durante un intervalo de tiempo corto.
- **Secuencia de salto de canal adaptado** es derivada de la BCHS y puede usar menos que las 79 frecuencias. Esta secuencia es usada en lugar de la BCHS.

2.2.2.5.1 Esquema de selección general.

Este consiste de 2 partes:

- Selección de secuencia;
- Mapeo de esta secuencia a saltos de frecuencia.

Las entradas de la caja de selección como se muestra en la Figura 2.6, son el reloj seleccionado, el reloj congelado, N , k_{offset} , dirección, selector de secuencia, y el AFH_channel_map. La fuente de reloj depende de la secuencia de salto seleccionada. Y cada secuencia usa diferentes bits del reloj.

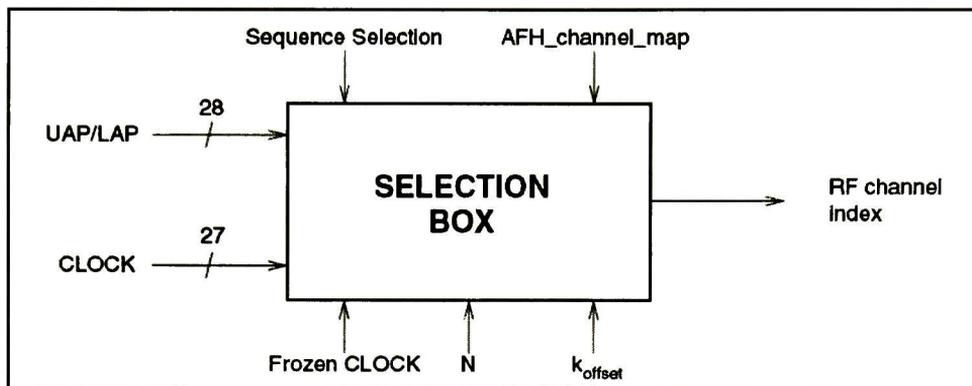


Figura 2.6 Diagrama a bloque del esquema de selección de salto [4].

La secuencia de selección puede ser cualquiera de los siguientes valores:

- Escaneo de page (PS).
- Escaneo de inquiry (IS).
- Page (PHS).
- Inquiry (IHS).
- Respuesta de page del maestro (MPRHS).
- Respuesta de page del esclavo (SPRHS).
- Respuesta de inquiry (IR)
- Canal básico (BCHS)
- Canal adaptado (ACHS).

La dirección de 28 bits incluye la LAP y los 4 LSB's de la UAP. La dirección BD_ADDR del maestro es usada en el caso de BCHS o ACHS. Para las secuencias de PHS, MPRHS, SPRHS, PSHS la dirección Bluetooth es dada por el Host del dispositivo paged. Cuando las secuencias de IHS, IR ó IS son seleccionadas la LAP/UAP correspondiente a la GIAC

Cuando se selecciona la ACHS, la entrada AFH_channel_map indica cuales canales pueden ser usados y cuáles no deben usarse.

El esquema de selección escoge un segmento de 32 frecuencias que abarca 64 MHz, y recorren estos 32 saltos en un orden pseudo-aleatorio. El ancho de banda de 64 MHz es debido al acomodo de las frecuencias; primero las pares del 0 al 78 y

después las impares de 1 al 79. Al terminarse ese segmento de 32 saltos, un nuevo segmento de 32 saltos es elegido, etc. En las secuencias de salto PHS, MPRHS, SPRHS, PS, IHS, IR y en IS, el mismo segmento de 32 saltos es usado todo el tiempo (el segmento es seleccionado por la dirección; diferentes dispositivos tienen diferentes segmentos de paging). En el estado CONNECTION, la secuencia se desliza por los 79 saltos como en la Figura 2.7. Este salto se mantiene fijo para toda la duración del paquete, sin importar si es de 1, 3 ó 5 ranuras de tiempo.

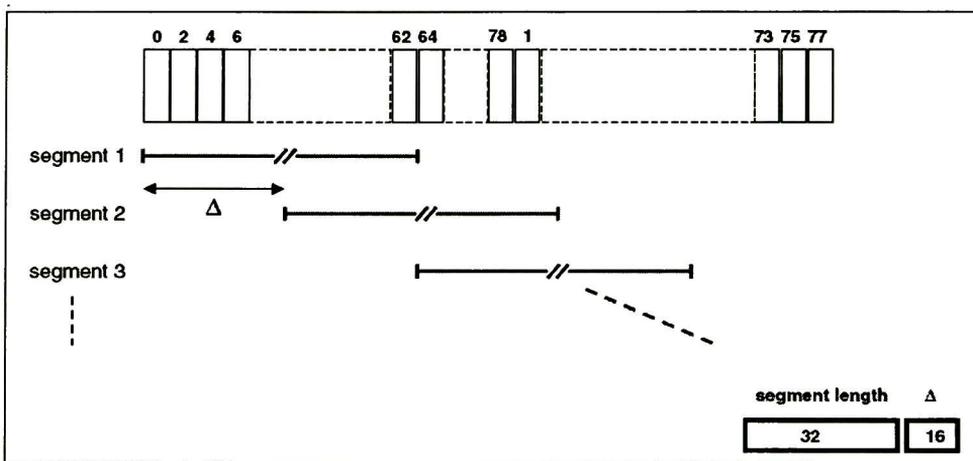


Figura 2.7 Esquema de selección de salto en el estado CONNECTION [4].

En el caso de ACHS, el esclavo responde con su paquete en el mismo canal de RF usado por el maestro para direccionar ese esclavo. Para mayor visualización del "mecanismo de mismo canal" se toma la Figura 2.8, k indica el TS ó ranura de tiempo, y f_k la frecuencia de transmisión ya sea del maestro al esclavo ó del esclavo al maestro. En la ranura $k+6$, el maestro transmite con una frecuencia f_{k+6} teniendo su paquete una duración de 5 TS, y en la ranura $k+11$, el esclavo contesta con la misma frecuencia que el maestro la f_{k+6} .

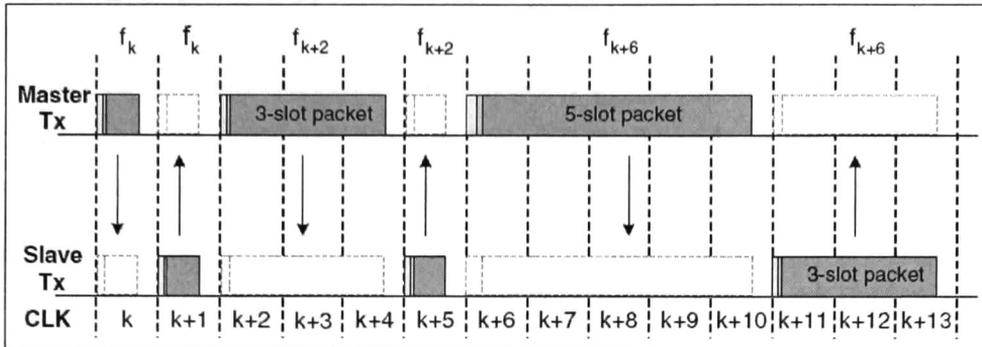


Figura 2.8 Ejemplo del mecanismo del mismo canal [4].

2.2.2.5.2 Kernel de selección.

La Figura 2.9 muestra el diagrama a bloques del kernel de selección de saltos básica, este kernel permite obtener las secuencias de PHS, PS, IHS, IR y la BCHS. La entrada X determina la fase dentro del segmento de 32 saltos, donde Y1 y Y2 seleccionan entre maestro a esclavo y esclavo a maestro. Las entradas A a D determinan el orden dentro del segmento, las entradas E y F determina el mapeo dentro de las frecuencias de salto. El kernel arroja una salida que sirve de direccionamiento a una memoria que contiene los canales ordenados, primero los canales pares y después los canales impares.

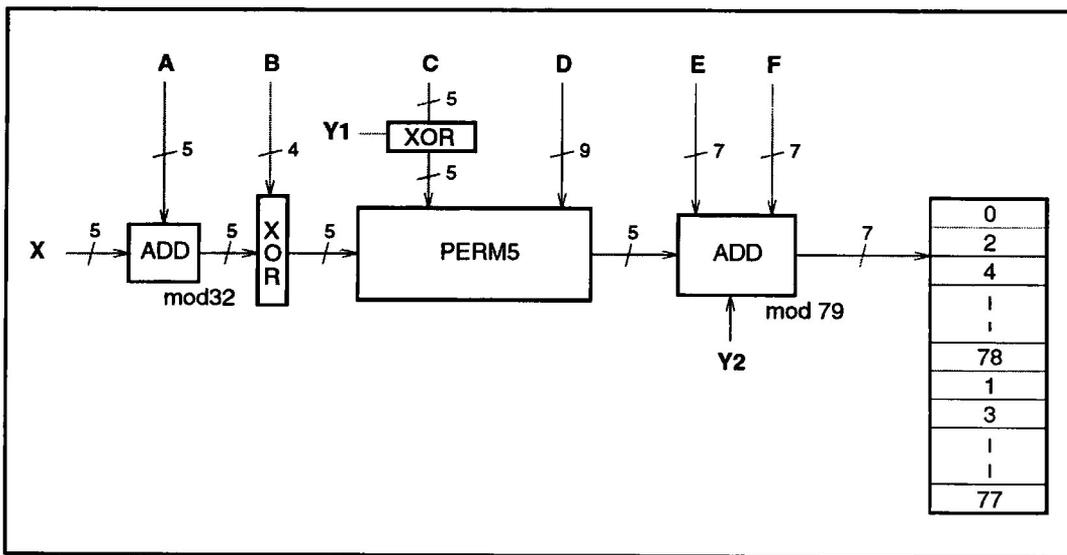


Figura 2.9 Diagrama a bloques del kernel de selección de saltos [4].

Operaciones:

- Primero se suma $A + X$ y al resultado se le aplica el modulo 32.
- Después, si al resultado del modulo 32 lo llamamos Z' y se realiza una operación XOR con B, donde B representa los bits A_{27-23} de la dirección Bluetooth, esta operación se ilustra en la Figura 2.10. Se hace notar que $Z'_4 = Z_4$

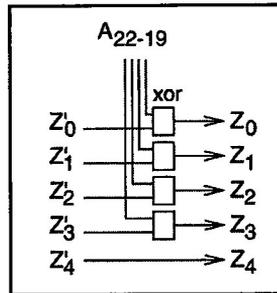


Figura 2.10 Operación XOR para el sistema de saltos [4].

- A continuación se hace una operación de permutación manipulada por una palabra de control. Esta permutación consiste de 7 etapas de operaciones de butterfly. Las señales de control se denominan P_{0-8} las cuales corresponden a las señales D_{0-8} , y las señales P_{9-13} al resultado de la operación de $C_{0-5} \oplus Y$. En la Tabla 2.4 Se muestra cuales señales de entrada Z's se ven afectados por la señal de control P.

Señal de control	Operación Butterfly	Señal de control	Operación mariposa
P_0	$\{Z_0, Z_1\}$	P_7	$\{Z_3, Z_4\}$
P_1	$\{Z_2, Z_3\}$	P_8	$\{Z_1, Z_4\}$
P_2	$\{Z_1, Z_2\}$	P_9	$\{Z_0, Z_3\}$
P_3	$\{Z_3, Z_4\}$	P_{10}	$\{Z_2, Z_4\}$
P_4	$\{Z_0, Z_4\}$	P_{11}	$\{Z_1, Z_3\}$
P_5	$\{Z_1, Z_3\}$	P_{12}	$\{Z_0, Z_3\}$
P_6	$\{Z_0, Z_2\}$	P_{13}	$\{Z_1, Z_2\}$

Tabla 2.4 Control de las butterflies para el sistema de saltos.

En la Figura 2.11, se ejemplifica una operación de butterfly para la señal de control P_0 .

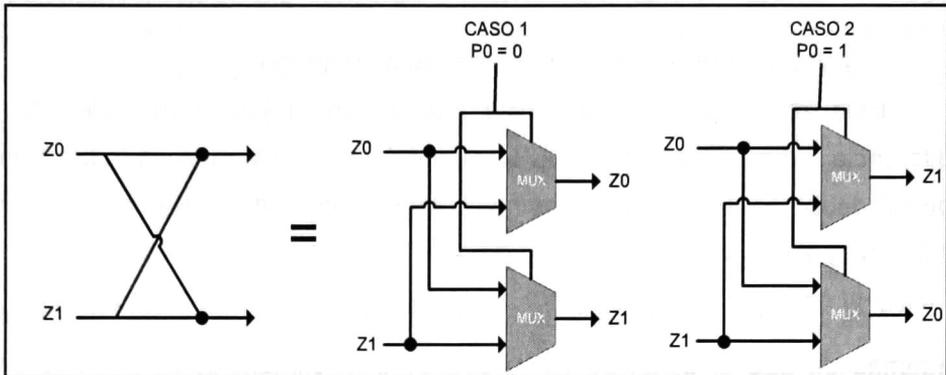


Figura 2.11 Operación Butterfly para el caso de P_0 .

En la Figura 2.12 se muestran etapas de permutación del sistema de saltos.

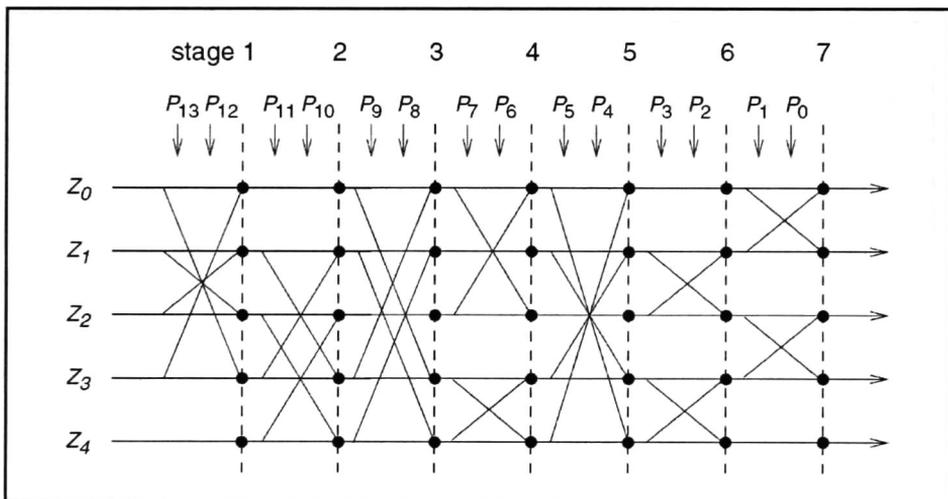


Figura 2.12 Operación de permutación para el sistema de salto [4].

- Una vez realizada la permutación, el resultado de 5 bits es sumado con los valores de E, F y Y2, el resultado de esta operación se le aplica una operación modulo 79.
- El resultado anterior es un dígito de 7 bits, por lo que permite direccionar una memoria de 79 registros. En este registro se guarda la palabra de

control para el sintetizador correspondiente a la frecuencia de salto entre 0 a 78.

2.2.2.5.3 Kernel de selección de salto adaptado.

Este kernel de selección está basado en el kernel de selección básico. La diferencia radica en la entrada AFH_channel_map. Esta entrada indica cuáles canales de RF deben ser usados y cuáles no. El número de canales N puede ser un valor entre $20 \leq N \leq 79$.

Los canales no usados son re-mapeados uniformemente en canales usados, además de que el canal de RF usado para la comunicación maestro-esclavo es el usado inmediatamente en la comunicación esclavo-maestro.

Un diagrama a bloques para el mecanismo de re-mapeo es mostrado en la Figura 2.13. Esta función es una etapa de post-procesamiento del kernel de selección.

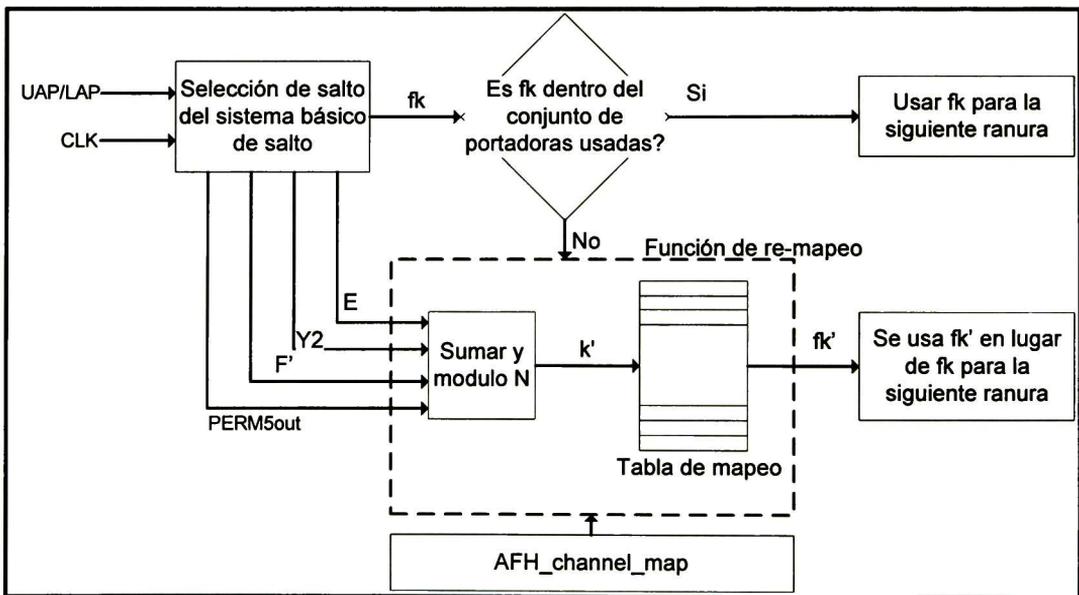


Figura 2.13 Diagrama a bloques del mecanismo de selección de salto adaptativo [4].

Cuando un canal no usado es generado por el mecanismo de selección básico, es re-mapeado a un conjunto de canales usados de la siguiente manera: un nuevo índice $k' \in \{0, 1, 2, \dots, N - 1\}$ es calculado usando algunos parámetros del kernel de selección de salto básico:

$$k' = (\text{PERM5}_{\text{out}} + E + F' + Y2) \bmod N, \quad \text{Ec. 2.1}$$

donde F' es definida en la Figura 2.14. El índice k' es usado direccionar una tabla de mapeo que contiene todos los canales RF pares usados en orden ascendente y después todos los canales RF impares usados en orden ascendente, los canales no permitidos o no usados son removidos de la tabla de mapeo.

2.2.2.5.4 Palabra de control.

Los valores de X, Y1, Y2, A a F que controla al kernel de selección de salto, varían dependiendo del estado en que se encuentre, los estados definidos en la Figura 2.14.

	Page scan / Interlaced Page Scan / Inquiry scan / Interlaced Inquiry Scan	Page/Inquiry	Master/Slave page response and Inquiry response	Connection state
X	$\text{CLKN}_{16-12} /$ $(\text{CLKN}_{16-12} + 16) \bmod 32 /$ $Xi_{4-0} /$ $(Xi_{4-0} + 16) \bmod 32$	Xp_{4-0} / Xi_{4-0}	$Xprm_{4-0} /$ $Xprs_{4-0} /$ Xir_{4-0}	CLK_{6-2}
Y1	0	$\text{CLKE}_1 / \text{CLKN}_1$	$\text{CLKE}_1 / \text{CLKN}_1 / 1$	CLK_1
Y2	0	$32 \times \text{CLKE}_1 /$ $32 \times \text{CLKN}_1$	$32 \times \text{CLKE}_1 /$ $32 \times \text{CLKN}_1 /$ 32×1	$32 \times \text{CLK}_1$
A	A_{27-23}	A_{27-23}	A_{27-23}	$A_{27-23} \oplus \text{CLK}_{25-21}$
B	A_{22-19}	A_{22-19}	A_{22-19}	A_{22-19}
C	$A_{8,6,4,2,0}$	$A_{8,6,4,2,0}$	$A_{8,6,4,2,0}$	$A_{8,6,4,2,0} \oplus \text{CLK}_{20-16}$
D	A_{18-10}	A_{18-10}	A_{18-10}	$A_{18-10} \oplus \text{CLK}_{15-7}$
E	$A_{13,11,9,7,5,3,1}$	$A_{13,11,9,7,5,3,1}$	$A_{13,11,9,7,5,3,1}$	$A_{13,11,9,7,5,3,1}$
F	0	0	0	$16 \times \text{CLK}_{27-7} \bmod 79$
F'	n/a	n/a	n/a	$16 \times \text{CLK}_{27-7} \bmod N$

Figura 2.14 Control para el sistema de saltos [4].

2.2.3 Enlace físico.

Un enlace físico representa una conexión banda base entre dispositivos. Un enlace físico es asociado con un canal físico. El enlace físico tiene propiedades comunes que aplican para todos los transportes lógicos sobre el enlace físico. Estas propiedades son:

- Control de potencia.
- Supervisión de enlace.
- Encriptación.
- Cambio de la velocidad de datos dependiendo de la calidad del canal.
- Control de paquetes multi-ranura.

2.2.3.1 Supervisión del enlace.

Una conexión puede romperse debido a varias razones como dispositivos moviéndose fuera del rango, interferencia severa ó falla en la fuente de poder. Dado que esto sucede sin alguna advertencia, es importante monitorear el enlace en ambos lados el maestro y el esclavo para evitar colisiones cuando la dirección de transporte lógico o dirección de miembro parked es reasignada a otro esclavo.

Para esta finalidad se utiliza un temporizador de supervisión de enlace, $T_{supervision}$. El temporizador debe ser reseteado por cada recepción de paquete, si esto no sucede y el temporizador alcanza el valor de "supervision TO" la conexión se considera desconectada.

El valor de $T_{supervision}$ es negociado por el administrador de enlace.

2.2.4 Transporte lógico.

Se tienen definidos en el estándar 5 tipos de transporte lógico:

- Transporte lógico orientado a conexión síncrono (SCO).
- Transporte lógico orientado a conexión síncrono extendido (eSCO).
- Transporte lógico orientado a conexión asíncrono (ACL).
- Transporte lógico de broadcast de esclavo activo (ASB.)
- Transporte lógico de broadcast de esclavo parked (PSB).

El transporte lógico síncronos SCO es un enlace simétrico y punto a punto entre maestro y un solo esclavo. Soporta voz o datos generales síncronos. La sincronía se obtiene mediante la reservación de ranuras de tiempo por parte del maestro, se puede considerar una conexión de conmutación de circuitos. El maestro puede soportar hasta 3 enlaces SCO, estos paquetes SCO nunca son retransmitidos.

En adición de las ranuras reservadas, el transporte lógico eSCO tiene ventana de retransmisión, puede ser simétrico o asimétrico.

El transporte lógico ACL es también punto a punto entre maestro y esclavo. Se envía información en las ranuras no reservadas, este transporte lógico es una conexión de conmutación de paquetes, entre maestro y esclavo solo un enlace ACL puede existir, en la mayoría de los paquetes ACL se utiliza la retroalimentación

El transporte lógico ASB es usado por un maestro para comunicarse con esclavos activos. Usa la dirección todos ceros en la dirección LT_ADDR.

EL transporte lógico PSB es usado por el maestro para comunicarse con los esclavos parked.

2.2.4.1 Dirección de transporte lógico (LT_ADDR).

A cada esclavo activo en la piconet se le asigna una dirección lógica de 3 bits (LT_ADDR). La dirección LT_ADDR con valor cero es reservada para mensajes de broadcast. Una LT_ADDR secundaria es asignada al esclavo por cada transporte lógico eSCO en uso en la piconet. La dirección LT_ADDR es transportada en el encabezado del paquete.

En el establecimiento de la conexión y en el intercambio de rol, la dirección primaria LT_ADDR es transportada por la carga útil del FHS.

2.2.4.2 Rutinas de Transmisión / Recepción.

Cada buffer síncrono consiste de 2 registros FIFO, un registro actual que puede ser leído por el Controlador del Enlace (LC) para ensamblar un paquete y un registro siguiente que puede ser leído por el Administrador de Recursos de Banda base (BRM) para cargar información. Los conmutadores son controlados por el LC. Los conmutadores nunca se conectan a un solo registro de manera simultánea.

La arquitectura de la recepción es de manera idéntica como en la Figura 2.15, solo que el flujo de la información es en sentido contrario.

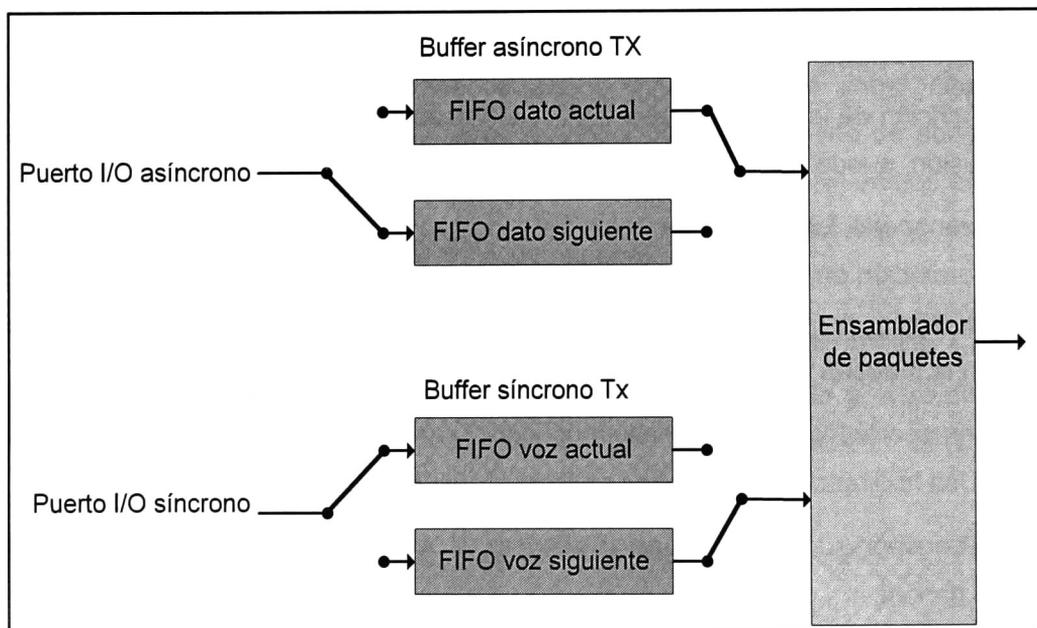


Figura 2.15 Diagrama funcional del buffer de transmisión.

2.2.5 Enlaces lógicos.

Se definen 5 enlaces lógicos:

- Control de enlace (LC).
- Control ACL (ACL-C).
- Usuario Asíncrono / Isócrono (ACL-U)
- Usuario Síncrono (SCO-S).
- Usuario Síncrono Extendido (eSCO-S).

El **enlace lógico de control LC** debe ser mapeado al encabezado del paquete. Este enlace transporta información de control de enlace de bajo nivel como el ARQ, control de flujo, y características del payload. El enlace lógico LC es transportado en cada paquete excepto en el paquete ID, el cual no tiene encabezado de paquete.

El **enlace lógico ACL-C** es usado para transportar información de control de enlace intercambiada entre los administradores de enlace. Este enlace usa paquetes DM1. Se indica en el encabezado del paquete en el campo LLID.

El **enlace lógico ACL-U** transporta datos de usuario isócrono y asíncrono L2CAP. Dado que este mensaje se puede enviar en varios paquetes de banda base es decir, mensajes fragmentados, el inicio del mensaje se indica con el código '10' en el campo LLID del encabezado de paquete, y los paquetes restantes usan el código '01'

El **enlace lógico SCO-S** transporta datos de usuario síncronos transparente. Este enlace es transportado sobre SCO.

El **enlace lógico eSCO-S** transporta datos de usuario síncrono transparente, este enlace es transportado por el eSCO.

2.2.6 Paquetes.

2.2.6.1 Formato general.

El formato general del paquete se muestra en la Figura 2.16, cada paquete consiste de **código de acceso, encabezado y payload**.

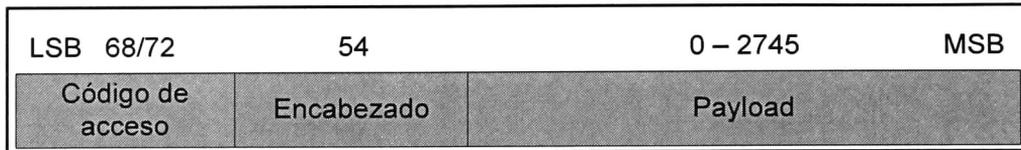


Figura 2.16 Formato del paquete general

El código de acceso es de 72 o 68 bits y el encabezado es de 54 bits. El rango del payload es de 0 a 2745 bits.

2.2.6.2 Orden de bits.

El ordenamiento de bits sigue el formato Little Endian. Esto significa:

- EL LSB corresponde a b_0
- EL LSB es el primero en enviarse sobre el aire.
- En las ilustraciones, el LSB es mostrado del lado izquierdo.

2.2.6.3 Código de acceso.

Cada paquete inicia con un código de acceso. Si el paquete tiene encabezado, el código de acceso es de 72 bits de longitud. En cualquier otro caso, el código de acceso es de 68 bits de longitud y es conocido como código de acceso recortado. Este código de acceso no contiene un trailer. El código de acceso es para sincronización, compensación de DC e identificación. En el receptor, un correlacionador deslizante correlaciona contra el código de acceso y se dispara cuando un umbral es excedido. Esta señal de disparo es usada para determinar el temporizado de recepción.

El código de acceso recortado es usado en los sub-estados de **paging, inquiry y park**, en este caso el código de acceso es usado como mensaje de señalización y no lleva encabezado ni payload.

El código de acceso consiste de preámbulo, palabra de sincronía, y posiblemente un trailer. En la Figura 2.17 se muestra la estructura del código de acceso.

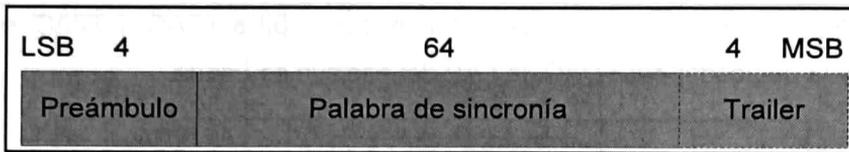


Figura 2.17 Formato del código de acceso.

Existen 4 tipos diferentes de códigos de acceso, resumidos en la Tabla 2.5. En el caso de DAC, GIAC y DIAC la longitud de 72 bits es usada en combinación con paquetes FHS.

Tipo de código	LAP	Longitud del código
CAC	Maestro	72
DAC	Dispositivo paged	68 / 72
GIAC	Reservado	68 / 72
DIAC	Dedicada	68 / 72

Tabla 2.5 Tipos de códigos de acceso

2.2.6.3.1 Preámbulo.

El preámbulo es un patrón fijo de 4 símbolos usados para facilitar la compensación de DC. La secuencia es 1010 o 0101 dependiendo del LSB de la palabra de sincronía es 1 ó 0, respectivamente.

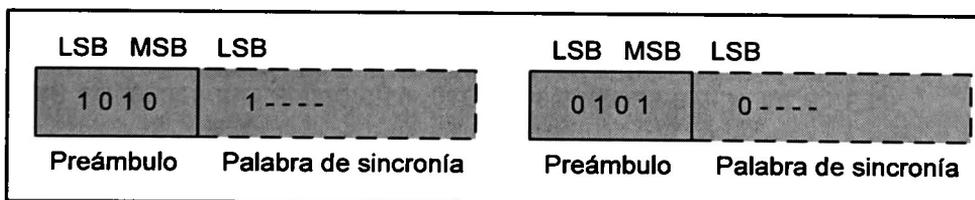


Tabla 2.6 Preámbulo

2.2.6.3.1 Palabra de sincronía.

La palabra de sincronía es una palabra de 64 bits derivada de los 24 bits de la dirección de LAP; el CAC es la LAP del maestro; para GIAC y DIAC es la LAP reservada y la dedicada; para DAC, la LAP del esclavo es usada.

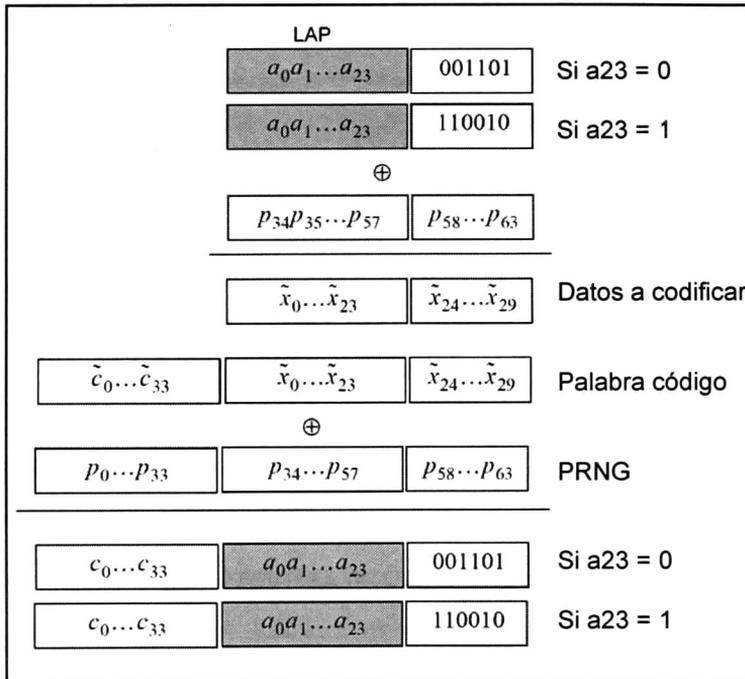


Figura 2.18 Contrucción de la palabra de sincronía [4].

La forma de construcción de la palabra de sincronía es mostrada en la Figura 2.18, se resume de la siguiente manera:

- Se consideran los 24 bits de la dirección LAP.
- Si el bit a_{23} es 0 se agrega una secuencia de 001101, en el caso que a_{23} es 1 se agrega la secuencia 110010, esto define una secuencia Barker de 7 bits, la cual ayuda a mejorar las propiedades de auto-correlación.
- El resultado se hace una XOR con los 30 últimos bits de la secuencia PRNG de las siglas en inglés *Pseudo-Random Noise Generator*.
- Esta palabra de 30 bits se pasa por un CRC de (64,30) el cual se explica más adelante, por lo que el resultado es una palabra de 64 bits.

- Los 64 bits de salida del CRC el cual se denomina palabra código, se pasan por una XOR con la secuencia de 64 bits generada por el PRNG definida en la Figura 2.19.

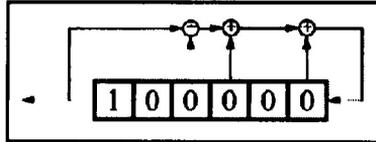


Figura 2.19 El LFSR y el estado inicial para generar la secuencia PRNG [4].

A la secuencia generada por el LFSR de las siglas en inglés *Linear Feedback Shift Register* se le agrega un cero para obtener la secuencia en notación hexadecimal es:

$$p = 3F2A33DD69B121C1 \quad \text{Ec. 2.2}$$

2.2.6.3.2 Trailer.

Dependiendo del tipo de código de acceso el trailer se puede o no usar, este es una patrón fijo de unos y ceros definidos en la Figura 2.20.

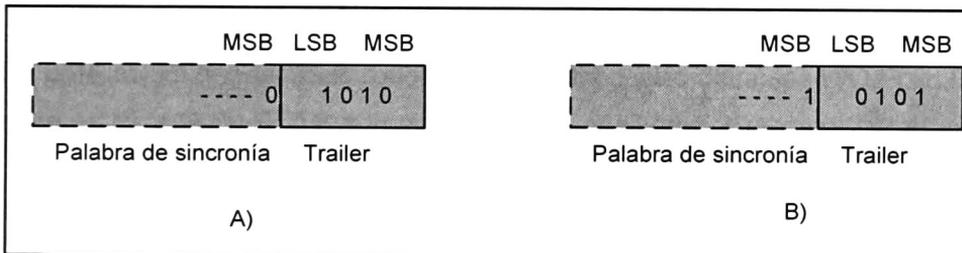


Figura 2.20 Trailer en CAC: a) MSB de la palabra de sincronía es 0, y b) MSB es 1.

Si el MSB de la palabra de sincronía es 0, se añade la secuencia 1010, en el caso de que sea 1 el MSB de la palabra de sincronía, se anexa la secuencia 0101.

2.2.6.4 Encabezado de paquete.

El encabezado contiene información de control de enlace (LC) y consiste de 6 campos, especificados en la Figura 2.21.

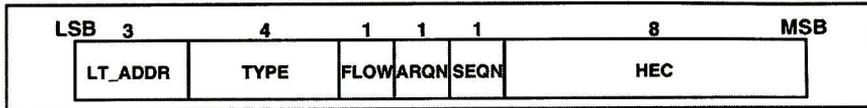


Figura 2.21 Formato del encabezado [4].

- LT_ADDR.

Es un campo de 3 bits que contiene la dirección de transporte lógico para el paquete. Este indica el esclavo destino en la transmisión maestro-esclavo e indica el esclavo fuente en la transmisión esclavo-maestro.

- TYPE.

Es un campo de 4 bits que permite distinguir entre los 16 diferentes tipos de paquete. Estos paquetes son definidos en la sección “Tipos de paquetes”

- FLOW.

El bit FLOW es usado para control de flujo sobre el transporte lógico ACL. Cuando el buffer de RX para ACL está lleno, una indicación de STOP (FLOW=0) se regresa al transmisor para parar temporalmente la transmisión.

- LT_ARQN.

El bit ARQN es una indicación de acknowledgement usada para informar a la fuente de una transferencia exitosa de datos de payload con CRC.

- SEQN.

El bit SEQN proporciona un esquema de numeración secuencial para ordenar el flujo de paquetes de datos.

- HEC.

Cada encabezado tiene un verificador de error de encabezado para verificar su integridad El HEC es una palabra de 8 bits, el generador de HEC es inicializado con

UAP en el paquete FHS en la respuesta de maestro, y con DCI en el paquete FHS en el inquiry response.

2.2.6.5 Tipos de paquetes.

Los paquetes usados en la piconet están relacionados con el transporte lógico usado. Para indicar el tipo de paquete se usan los 4 bits denominados TYPE del encabezado de paquete.

Segmento	TYPE $b_3b_2b_1b_0$	Ranuras ocupadas	Transporte Lógico SCO	Transporte Lógico eSCO	Transporte Lógico ACL
1	0000	1	NULL	NULL	NULL
	0001	1	POLL	POLL	POLL
	0010	1	FHS	RESERVADO	FHS
	0011	1	DM1	RESERVADO	DM1
2	0100	1	NO DEFINIDO	NO DEFINIDO	DH1
	0101	1	HV1	NO DEFINIDO	NO DEFINIDO
	0110	1	HV2	NO DEFINIDO	NO DEFINIDO
	0111	1	HV3	EV3	NO DEFINIDO
	1000	1	DV	NO DEFINIDO	NO DEFINIDO
	1001	1	NO DEFINIDO	NO DEFINIDO	AUX1
3	1010	3	NO DEFINIDO	NO DEFINIDO	DM3
	1011	3	NO DEFINIDO	NO DEFINIDO	DH3
	1100	3	NO DEFINIDO	EV4	NO DEFINIDO
	1101	3	NO DEFINIDO	EV5	NO DEFINIDO
4	1110	5	NO DEFINIDO	NO DEFINIDO	DM5
	1111	5	NO DEFINIDO	NO DEFINIDO	DH5

Tabla 2.7 Paquetes definidos para transportes lógicos síncronos y asíncronos

2.2.6.6 Formato de payload.

En el payload se distinguen 2 campos: el campo para datos síncronos y el campo para datos asíncronos. Los paquetes ACL solo tienen campo de datos asíncronos y los paquetes SCO y eSCO solo tienen campo de datos síncrono.

2.2.6.6.1 Campo de datos síncrono.

En los paquetes SCO, este campo es de longitud fija, y no tiene encabezado de payload. En los paquetes eSCO el campo consiste de 2 segmentos: cuerpo de datos síncrono y código CRC., de igual manera el encabezado no existe.

- 1) Cuerpo de datos síncronos: para paquetes HV y DV este valor es fijo. Para paquetes EV, la longitud es negociada durante el setup del LMP eSCO.
- 2) Código CRC: Los 16 bits de CRC en el payload es generada como se ve en la Figura 2.27.

2.2.6.6.2 Campo de datos asíncrono.

En el caso de paquetes ACL el campo de datos asíncronos consiste de 2 o 3 segmentos: encabezado de payload, cuerpo de payload y posiblemente un código CRC.

- 1) Encabezado de payload: es de 1 o 2 bytes de longitud. De la Tabla 2.7, los paquetes del segmento 1 y 2 tienen 1 byte de encabezado de payload como se ve en la Figura 2.22. Y los paquetes en segmentos 3 y 4 tienen 2 bytes de encabezado de payload, esto se visualiza en la Figura 2.23.

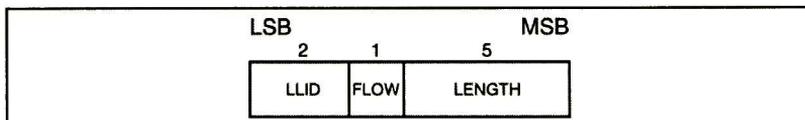


Figura 2.22 Encabezado de payload para paquetes ACL de una ranura [4].

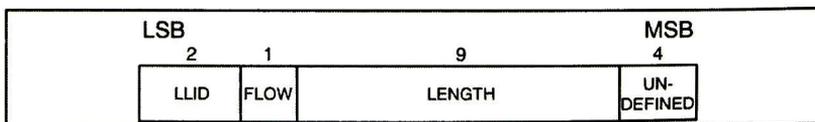


Figura 2.23 Encabezado de payload para paquetes ACL de multi-ranura [4].

- 2) Cuerpo de payload: incluye información de usuario y determina el throughput efectivo de usuario. La longitud del cuerpo es definida en la cabecera de payload.
- 3) Código CRC: Se usa un código de redundancia cíclica de 16 bits.

2.2.7 Procesamiento del flujo de bits.

Antes de que el payload sea enviado por el aire, algunas manipulaciones de bits son realizadas en el transmisor para incrementar la seguridad y la fiabilidad.

Al encabezado del paquete se le anexa un HEC para darle fiabilidad, después los datos son revueltos con un palabra de whitening, y al final una codificación FEC es aplicada, y en el receptor se realiza el proceso inverso; esto se muestra en la Figura 2.24.

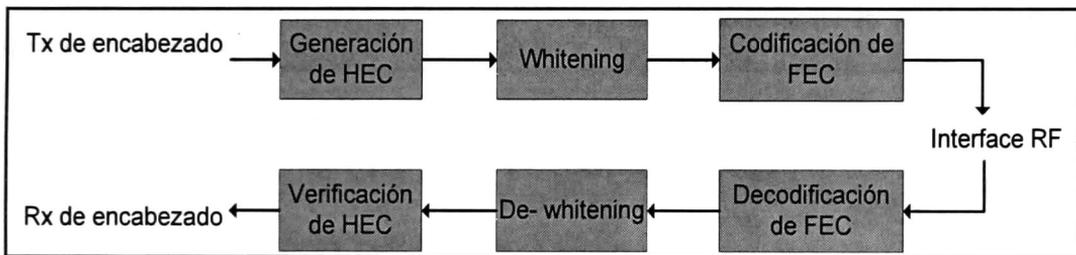


Figura 2.24 Procesamiento del encabezado

Para el payload, la generación de CRC, encriptación y codificación son opcionales; el whitening es obligatorio para el procesamiento del payload, esto se ejemplifica en la Figura 2.25, las líneas punteadas en los módulos indican que son opcionales para ciertos paquetes.

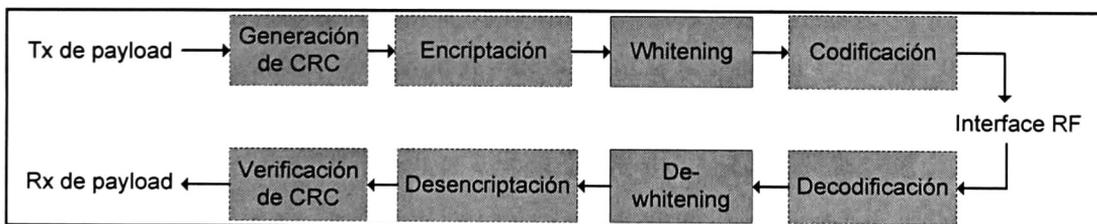


Figura 2.25 Procesamiento del payload.

2.2.7.1 Verificación de errores

La validación del código de acceso previene aceptar paquetes de otra piconet, siempre y cuando la LAP no sea la misma, si esto llegase a suceder el HEC y CRC se inicializan con la UAP del maestro, por lo que el HEC o CRC fallarían ya que viene de una piconet diferente.

2.2.7.1.1 Generación de HEC

EL HEC es un CRC generado por el polinomio

$$g(D) = D^8 + D^7 + D^5 + D^2 + D + 1 \quad \text{Ec. 2.3}$$

El circuito que realiza la operación de CRC es precargado con la UAP, inmediatamente el conmutador S es pasado a la posición 1, hasta que el último dato es introducido al LFSR el conmutador es puesto en la posición 2, y el HEC se lee del registro, esto se visualiza en la Figura 2.26.

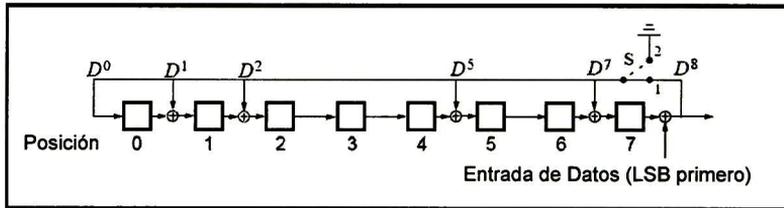


Figura 2.26 Circuito LFSR para la generación del HEC [4].

2.2.7.1.2 Generación de CRC

El LFSR de 16 bits para el CRC es construido de manera similar al de HEC usando el polinomio generador:

$$g(D) = D^{16} + D^{12} + D^5 + D^2 + 1 \quad \text{Ec. 2.4}$$

En este caso, los 8 bits a la izquierda son inicializados con la UAP mientras que los restantes 8 bits son puestos a cero.

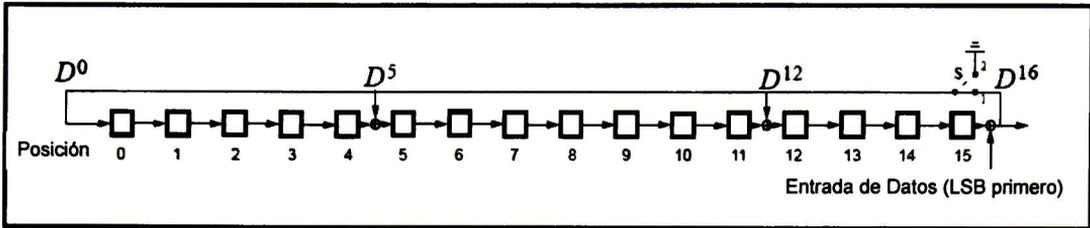


Figura 2.27 Circuito LFSR para la generación de CRC [4].

2.2.7.2 Whitening de datos.

Antes de la transmisión, el encabezado de paquete y el payload deben ser revueltos con una palabra de whitening para revolver los datos de patrones redundantes y minimizar el DC bias en el paquete. En el receptor los datos deben pasar por un de-whitening.

La palabra de whitening es generada por el polinomio $g(D) = D^7 + D^4 + 1$ y debe ser generada por un LFSR mostrado en la Figura 2.28. Antes de cada transmisión el registro debe ser inicializado con una porción del reloj del Bluetooth maestro CLK_{6-1} extendido con el MSB con un uno. Una excepción es el paquete FHS donde la inicialización es tomando el valor de la entrada X usada en la rutina de inquiry o respuesta de page, extendida con 1's en los MSB's.

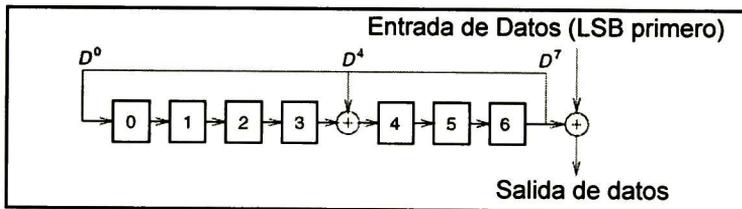


Figura 2.28 Circuito de LFSR del whitening de datos [4].

2.2.7.3 Corrección de errores.

Existen 3 esquemas de corrección de errores definidos en Bluetooth:

- FEC de 1 / 3.
- FEC de 2 / 3.
- Esquema ARQ para datos.

El objetivo del esquema FEC es reducir el número de retransmisiones. El encabezado de paquete es protegido por un FEC 1/3.

2.2.7.3.1 FEC 1 / 3.

Una simple repetición de 3 veces es usada para el encabezado. Esto se realiza repitiendo cada bit 3 veces, esta codificación es usada en el encabezado y en el campo de datos síncronos en el paquete **HV1**.

2.2.7.3.2 FEC 2 / 3.

Otro esquema FEC es un código Hamming recortado (15,10). El polinomio generador es un $g(D) = (D + 1)(D^4 + D + 1)$. El generador LFSR es mostrado en la Figura 2.29

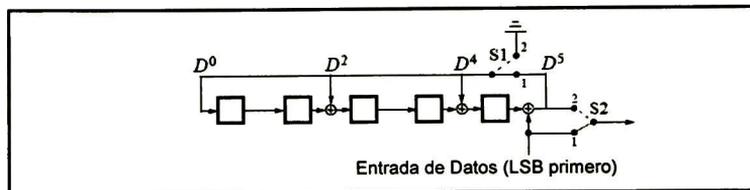


Figura 2.29 Generador LFSR para el código Hamming (15, 10) [4].

El funcionamiento es como sigue: los 10 bits son alimentados dentro del LFSR con los conmutadores S1 y S2 en la posición 1, después del último bit introducido los conmutadores se pasan a la posición 2, y los 5 bits de paridad son extraídos del registro. Este código puede corregir todos los posibles errores sencillos y detectar 2 errores en las palabras código. Este FEC es usado en los paquetes **DM**, en el campo de datos del paquete **DV**, en el paquete **FHS**, **HV2**, **EV4**.

2.2.7.3.3 Esquema ARQ.

De las siglas en inglés Automatic Repeat Request, es un esquema utilizado en los paquetes **DM**, **DH**, **DV** y **EV** el cual debe transmitir el dato hasta la recepción exitosa de un acknowledgement (ACK).

2.2.8 Operaciones del controlador de enlace.

Esta sección describe como se establece una piconet y como los dispositivos pueden anexarse y la liberación de la piconet, la Figura 2.30 es un diagrama de todos los estados y sub-estados del dispositivo Bluetooth.

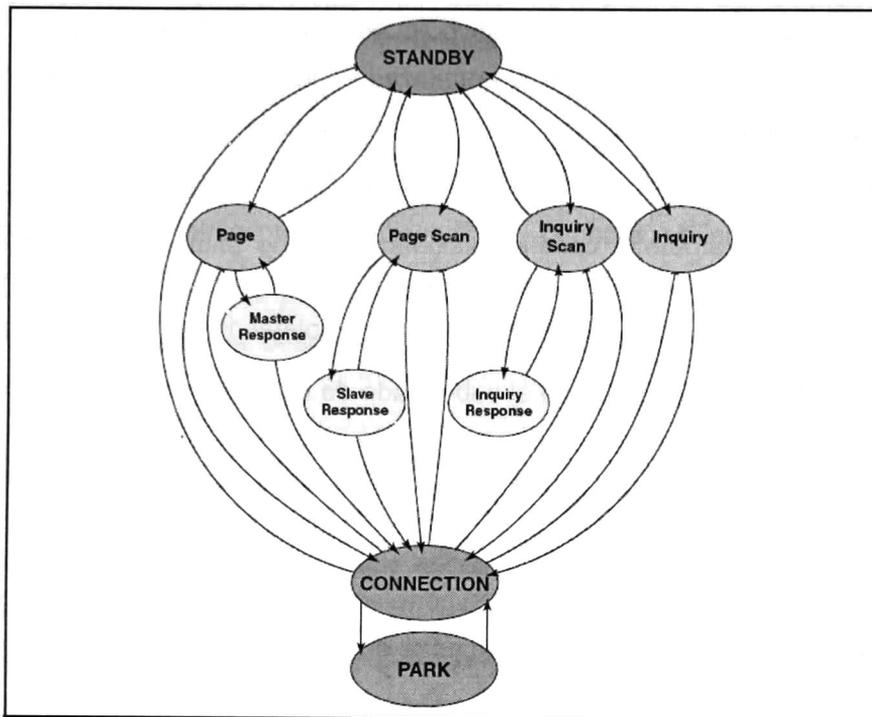


Figura 2.30 Diagrama de estados del controlador de enlace [4].

2.2.8.1 Estado *STANDBY*.

El estado *STANDBY* es el estado inicial del dispositivo Bluetooth. En este estado, el dispositivo puede estar en modo de bajo consumo. Solo el reloj nativo está corriendo.

El controlador puede dejar el estado *STANDBY* para escaneo de page o escaneo de mensajes de inquiry, o para page o inquiry el mismo.

2.2.8.2 *Sub-estados de establecimiento de conexión.*

Para establecer nuevas conexiones se usa el procedimiento de paging. Sólo se requiere la dirección de dispositivo Bluetooth para establecerla. Conocimiento acerca del reloj, obtenido del procedimiento de inquiry o de una conexión previa y el modo de

escaneo de page del otro dispositivo puede acelerar el procedimiento de establecimiento. El dispositivo que lleva a cabo el procedimiento de page se convierte en el maestro de la conexión.

2.2.8.2.1 Sub-estado de escaneo de page.

En este sub-estado, un dispositivo puede ser configurado para usar el procedimiento de escaneo estándar ó entrelazado. Durante un escaneo estándar un dispositivo escucha por la duración de una ventana de escaneo $T_{w_page_scan}$ (11.25 ms) mientras que en el escaneo entrelazado realizado como 2 escaneos de $T_{w_page_scan}$. Durante cada ventana de escaneo, el dispositivo escucha en una frecuencia, y el correlacionador iguala su código de acceso de dispositivo (DAC). La ventana de escaneo debe ser lo suficientemente grande para completar 16 frecuencias de page.

Cuando un dispositivo entra en el sub-estado de escaneo de page, selecciona las frecuencias de scan de acuerdo a la secuencia de saltos de page determinadas por la dirección del dispositivo Bluetooth. La fase en la secuencia es determinada por $CLKN_{16-12}$ donde cada 1.28 seg una frecuencia diferente es seleccionada.

En el caso de escaneo estándar, si el correlacionador excede el umbral de disparo durante el escaneo de page, el dispositivo entra en el estado de respuesta de esclavo.

2.2.8.2.2 Sub-estado de page.

Este sub-estado es usado por el maestro para activar y conectar a un esclavo en el sub-estado de **escaneo de page**. El maestro trata de coincidir con el escaneo del esclavo repitiendo la transmisión de mensajes de paging el cual consiste del código de acceso de dispositivo esclavo en diferentes canales. Dado que los relojes Bluetooth del maestro y el esclavo no están sincronizados, el maestro no conoce exactamente cuando el esclavo despierta y en cual salto de frecuencia. Entonces, el maestro transmite un tren de mensajes de escaneo de page idénticos en diferentes frecuencias de salto y escucha entre los intervalos de transmisión hasta que recibe una respuesta del esclavo.

El procedimiento de page en el maestro consiste de varias etapas. Primero el Host comunica la BD_ADDR del esclavo al Controlador. Esta BD_ADDR la usa el

maestro para determinar la secuencia de saltos de page. Para la fase en la secuencia, el maestro usa un estimado del reloj del esclavo. Este estimado puede ser derivado de una conexión pasada o de un procedimiento de inquiry.

La estimación del reloj Bluetooth en el esclavo puede estar errónea. Aunque las secuencias de salto sean las mismas del maestro y el esclavo, estos tienen diferentes fases y nunca podrían seleccionar la misma frecuencia. Para compensar esto, el maestro envía el mensaje de page durante intervalos de tiempo corto sobre frecuencias de wake-up. El maestro debe transmitir en frecuencias de saltos anteriores y posteriores a la frecuencia actual. Durante cada ranura de transmisión, el maestro secuencialmente transmite en 2 frecuencias de salto diferentes. Por lo tanto en la ranura de recepción escucha por esas 2 frecuencias diferentes por paquetes de ID. Si una respuesta es obtenida desde el esclavo, el maestro entra en el sub-estado de **respuesta de maestro**.

2.2.8.2.3 Sub-estado de respuesta de page.

Cuando un mensaje de page es exitosamente recibido por el esclavo, existe una sincronía burda de frecuencias de salto entre el maestro y el esclavo. El maestro y el esclavo entran en el sub-estado de respuesta para intercambiar información vital para continuar con el establecimiento de la conexión. Es imprescindible que los dos dispositivos usen el mismo CAC, la misma secuencia de saltos de canal y los relojes estén sincronizados, estos datos se obtienen del dispositivo maestro. Y son enviados en este sub-estado.

En el paso 1, el maestro está en el sub-estado de **page** y el esclavo está en el sub-estado de **escaneo de page**, si el mensaje es recibido por el esclavo, este entra en sub-estado de **respuesta de esclavo** en el paso 2. El maestro espera por una contestación del esclavo y cuando esta llega, el maestro entra en **respuesta de maestro** en la etapa 3. Los parámetros son derivados de la BD_ADDR del esclavo.

Si el esclavo recibe un paquete **FHS** en el sub-estado de **respuesta de esclavo**, este regresa un mensaje de respuesta de page de esclavo en la etapa 4 para confirmar la recepción del paquete de FHS. En este caso el esclavo debe cambiar al CAC del maestro y al reloj del maestro recibido en el paquete FHS.

Con esto el esclavo entra en el estado **CONNECTION** en el paso 5. El modo de conexión inicia con un paquete de **POLL** transmitido por el maestro. Y el esclavo debe contestar con cualquier tipo de paquete. Un resumen de estos pasos se muestra en la **Tabla 2.8**

Paso	Mensaje	Tipo de paquete	Dirección	Secuencia de salto	Código de acceso y reloj
1	Page	ID	Maestro a esclavo	Page	Esclavo
2	1º respuesta de page del esclavo	ID	Esclavo a maestro	Respuesta de page	Esclavo
3	Respuesta del maestro	FHS	Maestro a esclavo	Page	Esclavo
4	2ª respuesta de page del esclavo	ID	Esclavo a maestro	Respuesta de page	Esclavo
5	1er paquete del maestro	POLL	Maestro a esclavo	Canal	Maestro
6	1er paquete del esclavo	Cualquiera	Esclavo a maestro	Canal	Maestro

Tabla 2.8 Mensajes iniciales durante el establecimiento de la conexión.

2.2.8.3 Sub-estados de descubrimiento de dispositivo.

Para descubrir otros dispositivos, el dispositivo Bluetooth debe entrar en el sub-estado de **inquiry**, en este se transmite de manera repetitiva un mensaje de **inquiry** (paquete ID) en diferentes frecuencias. La secuencia de salto del sub-estado de **inquiry** es derivada de la LAP de la GIAC. Incluso cuando las DIAC's son usadas, la secuencia de saltos es generada de la LAP de la GIAC. Un dispositivo que permite ser descubierto, entra en el sub-estado de **escaneo de inquiry** para responder a mensajes de **inquiry**.

Durante el sub-estado de **inquiry**, el dispositivo descubridor colecciona las direcciones de dispositivo Bluetooth y el reloj de todos los dispositivos que responden

al mensaje de inquiry. Entonces, si lo desea, puede hacer una conexión con alguno de ellos previo procedimiento de page.

El mensaje difundido de inquiry por la fuente no contiene ninguna información, ya que solo hay un GIAC para inquiry.

2.2.8.3.1 Sub-estados de escaneo de inquiry.

El sub-estado de **escaneo de inquiry** es similar al escaneo de page. En lugar de escanear por el código de acceso del dispositivo, el receptor escanea por el código de acceso de inquiry. De igual manera existen 2 tipos de escaneo: estándar y entrelazado. La secuencia es determinada por el GIAC y la fase por el reloj nativo, la fase cambia cada 1.28 segundos.

La entrada a este estado puede llegarse desde el estado STANDBY ó CONNECTION. El intervalo de escaneo es el tiempo entre 2 escaneos de inquiry consecutivos definido como $T_{\text{inquiry scan}}$ y debe ser menor a 2.56 segundos.

2.2.8.3.2 Sub-estados de Inquiry.

Este sub-estado es usado para descubrir nuevos dispositivos. Las secuencias de salto de frecuencias son determinadas por el GIAC y el reloj nativo del dispositivo descubridor. Entre transmisiones, el receptor escucha por mensajes de respuesta de inquiry. Este mensaje es un paquete FHS, el mensaje de respuesta no es confirmado de recibido. Este sub-estado es parado por el Administrador de Recursos de Bandabase (BRM).

2.2.8.3.3 Sub-estados de respuesta de inquiry.

En el primer mensaje de inquiry recibido en este sub-estado de escaneo de inquiry, el receptor debe regresar un paquete de respuesta de inquiry (FSH) el cual contiene el BD_ADDR y otros parámetros, y debe ser enviado 625 μseg después de que el mensaje de inquiry fue recibido.

Los mensajes enviados durante la rutina de inquiry es resumida en la Tabla 2.9. Donde el maestro envía mensajes de inquiry usando el GIAC y su reloj. Y el esclavo responde con un FHS.

Paso	Mensaje	Tipo de paquete	Dirección	Secuencia de salto	Código de acceso y reloj
1	Inquiry	ID	Maestro a esclavo	Inquiry	Inquiry
2	Respuesta de inquiry	FHS	Esclavo a maestro	Respuesta de inquiry	Inquiry

Tabla 2.9 Mensajeo durante la rutina de inquiry.

2.2.8.4 Estado CONNECTION

En este estado la conexión fue establecida y se puede mandar paquetes en cualquier dirección, se usa la secuencia básica o adaptada. La comunicación inicia con un paquete de POLL para el maestro para verificar la conmutación al temporizado del maestro y su secuencia de saltos de frecuencia.

Este estado es dejado con un comando de **detach o reset**.

En este estado de CONNECTION el dispositivo puede estar en alguno de los diferentes sub-estado:

- **Modo Activo:** En este estado el maestro y el esclavo participan en el canal. Los esclavos escuchan en la ranura de transmisión de maestro-esclavo.
- **Modo Sniff:** la duración del ciclo de la actividad del esclavo es reducida
- **Modo Hold:** en este modo, el esclavo no soporta paquetes ACL temporalmente, pero sí paquetes síncronos y los transmite en las ranuras reservadas. La capacidad del dispositivo se libera y puede hacer otras cosas como scanning, paging, inquiring o atender otra piconet.

2.2.8.5 Estado PARK.

Cuando un esclavo no necesita participar en un canal de piconet pero necesita permanecer sincronizado al canal, entra en el estado de PARK, este es un estado con poca actividad, el esclavo renuncia a la dirección de transporte LT_ADDR, pero recibe 2 nuevas direcciones. PM_ADDR y AR_ADDR de 8 bits de longitud.

CAPÍTULO 3 DESCRIPCIÓN DEL MODELO DE ORO.

3.1 Introducción

En este capítulo se realizará una descripción del modelo de oro implementado en MATLAB para la construcción de la trama Bluetooth y la generación de los 16 tipos diferentes de paquetes Bluetooth, así mismo, se plantea la arquitectura general para el dispositivo controlador. Para la formación de paquetes se realizó: la programación para la codificación de canal, generación de código de acceso de canal, encabezado de paquete, y sobre todo el kernel de selección de saltos. Las pruebas y validación de los datos generados se especifican en el Capítulo 4.

El modulador / demodulador GMSK se considera dentro del procesamiento banda base, por lo que se hizo un análisis de las arquitecturas encontradas en la literatura especializada, así como la justificación de la elección de una de ellas para usarse en el proyecto de Bluetooth. De la misma manera, se analizará las arquitecturas para demoduladores GMSK y la justificación del porque se propone usar una de estas arquitecturas. Cabe hacer mención que la información respecto al demodulador GMSK se encontró muy dispersa dentro de la bibliografía especializada, o las arquitecturas de los demoduladores GMSK se presentaba como caja negra, debido a que esto representa propiedad intelectual de los desarrolladores y/o investigadores, y es información que se guarda con recelo.

3.2 Arquitectura del procesador banda base.

En la Figura 3.1 se muestra el top level de la propuesta para el procesador banda base, este permite la entrada de flujos de datos SCO (Voz) y ACL (Datos), las señales de control y la entrada de reloj; en las salidas se consideran la salida de datos modulados en GMSK; la recepción de datos se espera que esté en banda base, por medio de la representación compleja pasa bajas de la señal GMSK, existe una señal de retroalimentación al front end para control de ganancia, y una señal la cual determina el DC offset; para dispositivos clase 1 y clase 2, es opcional controlar la potencia de salida del transceptor, se anexó una salida para poder tener este control; el frequency hopping se calcula en el procesador banda base y este valor se alimenta al front end, para que el sintetizador realice la sintonización a la frecuencia de salto, y una salida de referencia de reloj.

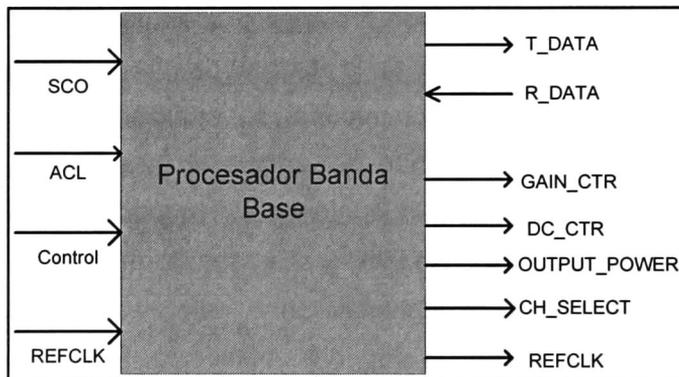


Figura 3.1 Top level del procesador banda base.

El segundo nivel de la arquitectura es mostrada en la Figura 3.2, revela los bloques internos del procesador banda base, en esta propuesta se pueden observar los siguientes bloques:

- Máquina de control de enlace.
- Buffer o memoria de puerto dual.
- Generador de palabra de sincronía.
- Generador de clave de encriptación.
- Control de Timing y CLK offset.

- Máquina de selección de saltos.
- Sincronía de símbolo y trama.
- Control de extracción de DC & ganancia.
- Ensamblador de paquetes.
- Modulador GMSK.
- Demodulador GMSK.

Los bloques de modulación y demodulación de la señal GMSK se describen a detalle en la sección 3.3.

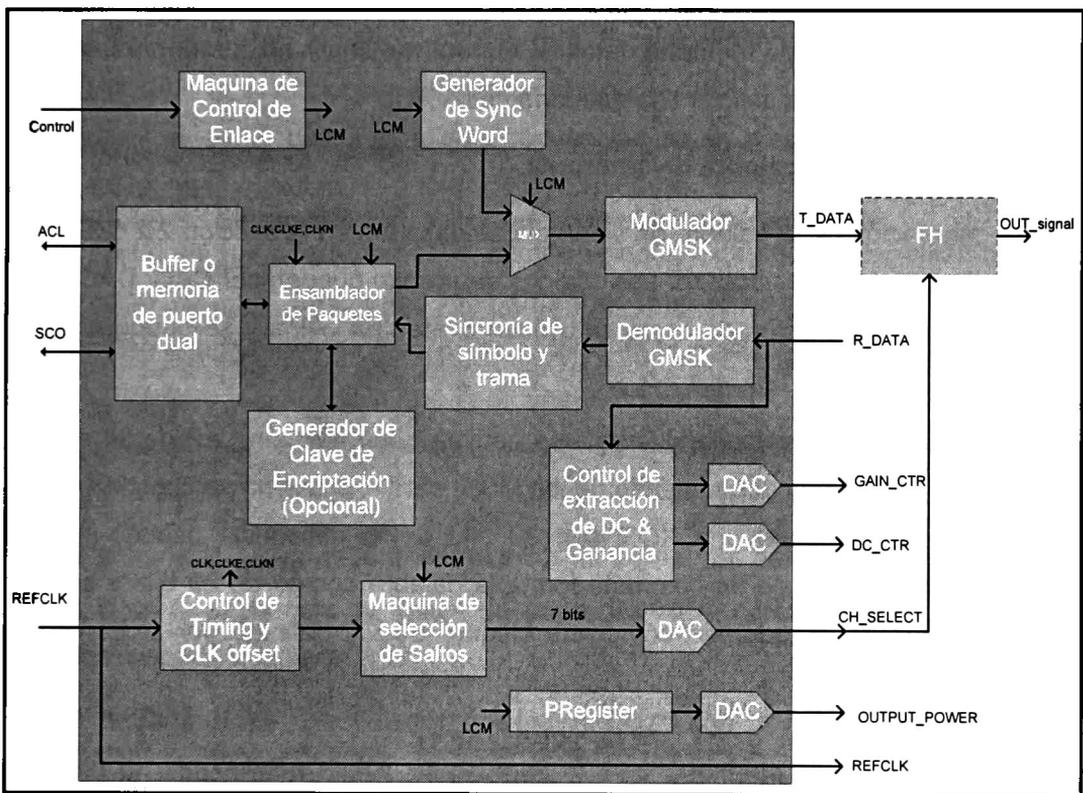


Figura 3.2 Bloques internos del procesador banda base.

3.2.1 Máquina de control de enlace.

La máquina de estados controla el procesador banda base por medio de la implementación del diagrama de estados de la Figura 2.30, la máquina alimenta ciertas señales al generador de palabra de sincronía, de igual manera a la máquina de

selección de saltos y al ensamblador de paquetes, además permite fijar la potencia de transmisión del dispositivo Bluetooth por medio del registro Pregister.

3.2.2 Buffer o memoria de puerto dual.

Esta memoria recibe y/o entrega información del transporte lógico orientado a conexión síncrono (SCO y eSCO), este transporte lógico se emplea principalmente para transferir voz en formato digital; el transporte lógico orientado a conexión asíncrona, usado para datos, estos flujos son administrados por el BRM.

El ensamblador de paquetes, dependiendo de sí las ranuras de tiempo están ó no reservadas, solicita a esta memoria datos SCO para transmitir en las ranuras, y en las ranuras no reservadas le solicita a la memoria datos ACL.

3.2.3 Generador de palabra de sincronía.

La generación de la palabra de sincronía se describe en la sección 2.2.6.3.1, esta forma parte del código de acceso. Los códigos de acceso pueden ser CAC, DAC, GIAC ó DIAC, para la generación de estos se utilizan diferentes LAP's y se puede anexar o no el trailer.

En resumen la palabra de sincronía se deriva de la dirección LAP de 24 bits a la cual se le agregan 6 bits dependiendo del bit MSB, para formar una secuencia Baker, a estos 30 bits se le hace una XOR con 30 bits de la secuencia pseudo-aleatoria generada por el PRNG de 64 bits, esta forma la palabra código de 30 bits a codificar por un polinomio generador de código de bloque (64,30) definido por la Ec. 3.1 en formato octal:

$$g(D) = 260534236651 \qquad \text{Ec. 3.1}$$

Por medio de un código de redundancia cíclica se generaran los 34 bits redundantes, la palabra código de 64 bits, se vuelve a pasar por una xor con la secuencia completa pseudo-aleatoria de 64 bits. Esto genera la palabra de sincronía completa a la cual si se le anexa el preámbulo y/o el trailer forma alguno de los códigos de acceso.

3.2.4 Control de timing & CLK offset.

El reloj Bluetooth debe tener 5 salidas de reloj, que son disparadores de eventos en el dispositivo, estas salidas se implementan con divisores de frecuencia; el CLK offset se modeló por medio de un sumador, este hace el ajuste entre el reloj maestro y el del esclavo. El valor de corrección es obtenido de un paquete ID. La Figura 2.2 muestra las salidas del reloj.

3.2.5 Máquina de selección de saltos.

Esta máquina de selección implementa el kernel descrito en la sección 2.2.2.5.2, este permite determinar el salto de frecuencia, los valores introducidos a este módulo son la UAP/LAP, la selección de secuencia, el reloj, la señal de reloj congelada, el valor N , k_{offset} y la señal que indica que frecuencias están disponibles en el salto de frecuencia adaptativo (AFH).

Los valores que puede tomar la entrada selección de secuencia, están definidas en la sección 2.2.2.5.1. La entrada **AFH_Channel_map** es un campo de 79 bits, donde los bits están enumerados de 0 a 78, un 1 lógico en la posición n , indica que ese canal es usado, un cero lógico indica que ese canal n no es usado, el rango de valores que puede tomar esta entrada van de $0x000000000000000000000000$ a $0x7FFFFFFFFFFFFFFFFFFFFFFF$.

La entrada k_{offset} puede tomar el valor de 24 u 8; 24 define la serie A y 8 define la serie B en la secuencia de saltos de page. Se inicia con la serie A, y cada 1.28 segundos se cambia la serie.

La entrada N es un contador que se inicializa con 0 y cada vez que el reloj Bluetooth en la posición CLK1 se pone en 0 se incrementa el contador en 1, este coincide con el inicio de una ranura de transmisión, esta entrada N se utiliza principalmente para las respuestas de page de esclavo y de maestro; y la respuesta de inquiry.

Cuando el dispositivo entra en respuesta de page, en el maestro o en el esclavo, se debe de introducir el valor del **reloj congelado**, este valor se obtiene cuando el paquete de ID es recibido.

3.2.6 Sincronía de trama.

La sincronía de trama se realiza por medio de una correlación entre los 64 bits de la palabra de sincronía dentro de código de acceso de canal y la secuencia de la señal recibida estimada. Representando la palabra de sincronía como:

$$\{x[m]\}_{0 \leq m \leq 63} = \{x_{63}, x_{62}, \dots, x_1, x_0\} \quad \text{Ec. 3.2}$$

Si se representa la secuencia de la señal recibida como:

$$\{\hat{x}[k - m.R]\}_{0 \leq m \leq 63} = \{\hat{x}[k], \hat{x}[k - R], \dots, \hat{x}[k - 63.R]\} \quad \text{Ec. 3.3}$$

Donde R denota el factor de sobremuestreo, la correlación se puede calcular como la propone [10] por medio de:

$$d[k] = \sum_{m=0}^{63} x[m] \oplus \hat{x}[k - m.R] \quad \text{Ec. 3.4}$$

Donde \oplus denota una suma modulo 2 realizada por medio de una operación XOR, para definir el inicio de la sincronía se define una bandera por medio de:

$$f[k] = \begin{cases} 1, & d[k] \leq d_{max} \\ 0, & d[k] > d_{max} \end{cases} \quad \text{Ec. 3.5}$$

Esto correspondé al número de errores $d[k]$, el cual es comparado con un número programable máximo de errores permitidos d_{max} .

3.2.7 Control de extracción de DC & ganancia.

Este módulo permite extraer el valor de DC en la secuencia de entrada, para posteriormente hacer una compensación del valor de DC, el control de ganancia permite retroalimentar un valor de RSSI (Received Signal Strength Indicator), esto permitirá al amplificador de entrada modificar su comportamiento, ya sea que incremente la amplificación porque la señal es muy débil ó disminuya la amplificación por que la señal se está saturando.

3.2.8 Ensamblador de paquetes.

El ensamblador de paquete mostrado en la Figura 3.3, articula los datos del paquete de acuerdo a la solicitud del Link Control Machine, para formar alguno de los

16 paquetes diferentes del estándar Bluetooth, la máquina de control de Tx /Rx genera ciertas señales para habilitar los bloques de CRC y/o el FEC, y el tipo de CRC y/o FEC a generar. He aquí la razón por la que estos bloques aparecen con líneas punteadas como opcionales. El módulo de whitening es obligatorio para todos los paquetes por lo que se dibuja con línea continua.

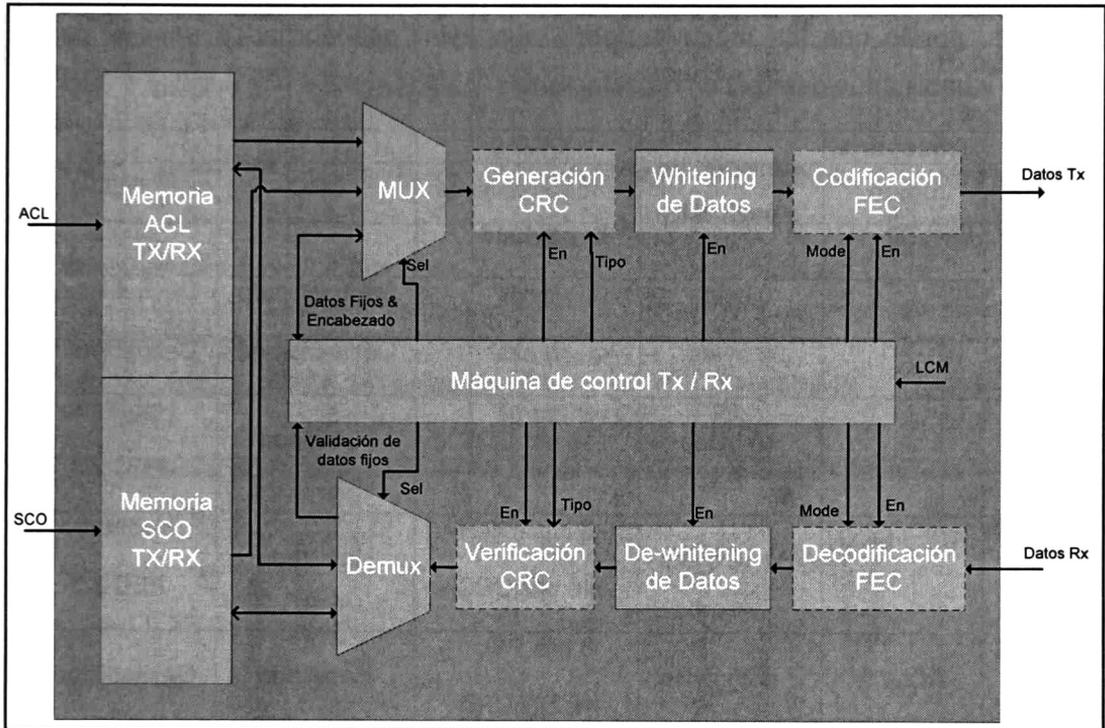


Figura 3.3 Diagrama a bloques del ensamblador de paquetes.

El diagrama de la Figura 3.3, anexa las memorias a manera de visualización, una de las entradas principales del módulo es el valor TYPE, el cual se encuentra en el encabezado del paquete.

Todos los paquetes tienen 18 bits de encabezado de paquete, como se muestra en la Figura 2.21, de estos 18 bits, 10 son datos y 8 bits son HEC; estos últimos son generados por un código de redundancia cíclica, especificado en la **Error! Reference source not found.**, el HEC se produce en el bloque de **Generación de CRC**. Los 18 bits pasan por un whitening constituido por un LFSR, y posteriormente son codificados

con un FEC de 1/3 lo que produce 54 bits de encabezado de paquete, como se muestra en la Figura 2.16.

La Tabla 3.1 muestra la señal o señales de habilitación que debe entregar la máquina de control Tx / Rx, y el tipo de FEC a generar dependiendo del paquete. Un caso especial es el paquete DV, en el cual los campos síncronos y asíncronos están ocupados, por lo que los módulos tienen que estar deshabilitados para el campo síncrono y habilitados para el campo asíncrono.

TYPE	Transporte Lógico / TS	Longitud de payload	CRC y encabezado de payload	CRC	FEC
NULL		0 bytes	0 bits	Deshabilitado	Deshabilitado
POLL		0 bytes	0 bits	Deshabilitado	Deshabilitado
FHS		18 bytes	16 bits CRC	Habilitado	Habilitado FEC 2/3
DM1	ACL / SCO / 1	1 a 18 bytes	8 bits PYLH 16 bits CRC	Habilitado	Habilitado FEC 2/3
DH1	ACL / 1	1 a 28 bytes	8 bits PYLH 16 bits CRC	Habilitado	Deshabilitado
HV1	SCO / 1	10 bytes	0 bits	Deshabilitado	Habilitado FEC 1/3
HV2	SCO / 1	20 bytes	0 bits	Deshabilitado	Habilitado FEC 2/3
HV3	SCO / 1	30 bytes	0 bits	Deshabilitado	Deshabilitado
DV	SCO / 1	10 bytes voz	0 bits	Deshabilitado	Deshabilitado
		1 a 10 bytes en datos	8 bits PYLH 16 bits CRC	Habilitado	Habilitado FEC 2/3

AUX	ACL / 1	1 a 30 bytes	8 bits PYLH	Deshabilitado	Deshabilitado
DM3	ACL / 3	2 a 123 bytes	16 bits PYLH 16 bits de CRC	Habilitado	Habilitado FEC 2/3
DH3	ACL / 3	2 a 185 bytes	16 bits PYLH 16 bits de CRC	Habilitado	Deshabilitado
EV4	eSCO / 3	1 a 120 bytes	16 bits de CRC	Habilitado	Habilitado FEC 2/3
EV5	eSCO / 3	1 a 180 bytes	16 bits de CRC	Habilitado	Deshabilitado
DM5	ACL / 5	2 a 226 bytes	16 bits PYLH 16 bits de CRC	Habilitado	Habilitado FEC 2/3
DH5	ACL / 5	2 a 341 bytes	16 bits PYLH 16 bits de CRC	Habilitado	Deshabilitado

Tabla 3.1 Tabla comparativa de los diferentes paquetes y señales de habilitación

3.3 Modulación / demodulación de la señal de fase continua digital GMSK.

3.3.1 Características de la modulación GMSK.

Las características del modulador GMSK proporcionadas por la especificación Bluetooth V1.2, son que el producto de ancho de banda por periodo de bit (BT) es de 0.5, y el índice de modulación debe estar entre 0.28 a 0.35. Este esquema de modulación digital fue introducido por Murota en [11]. De donde se toman las propiedades más importantes de este esquema de modulación:

- Ancho de banda angosto: esto suprime las componentes de alta frecuencia.
- Eficiencia en el espectro de potencia de salida: con un $BT = 0.5$. Esto concentra el 90% de la potencia en 0.69 tomando un ancho de banda normalizado. En la Figura 3.4 se muestra esta característica de la modulación GMSK. Esta eficiencia espectral genera interferencia intersimbólica en el transmisor.
- La interferencia al canal adyacente se minimiza, ya que la radiación de potencia fuera de banda es mínima.

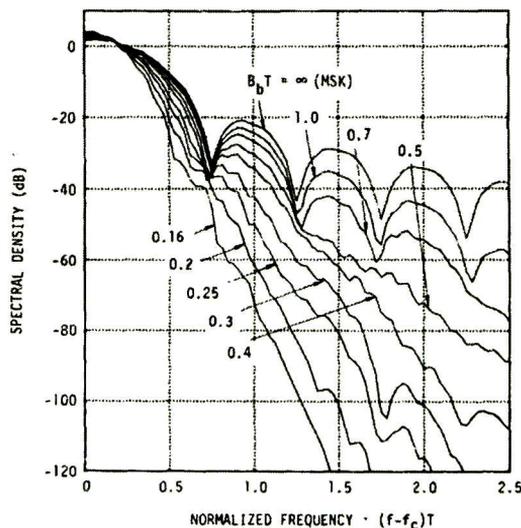


Figura 3.4 Espectro de densidad de potencia de GMSK [11].

Otro dato importante es la curva de BER obtenida de este artículo, la cual muestra un valor aproximado de E_b/N_0 de 9 dB para una relación señal a ruido de 10^{-3} con un BT 0.25. La Figura 3.5 muestra las curvas de BER para diferentes valores de producto de ancho de banda por periodo de bit (BT).

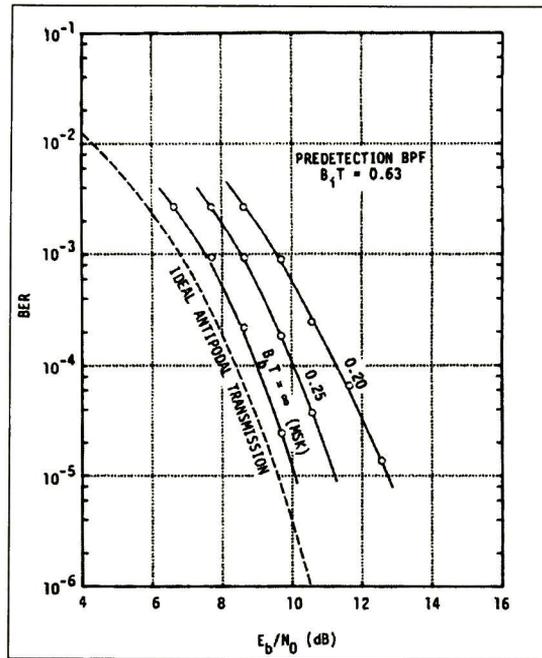


Figura 3.5 Desempeño de BER en un canal AWGN [11].

La generación de la señal GMSK en este artículo se realiza al pasar los datos binarios NRZ por un shaping de filtro Gaussiano, y a continuación por un modulador FM (VCO), en el demodulador se utilizó un detector coherente cuyas partes principales son: la recuperación de portadora por un Lazo de Costas y un duplicador de frecuencia.

3.3.1.1 Representación matemática de la señal GMSK.

La expresión general de las señales que hacen uso de la fase para transmitir información es de la forma:

$$s(t) = \left(\frac{2E}{T}\right)^{\frac{1}{2}} \cos(w_0 t + \phi(t)) \quad \text{Ec. 3.6}$$

Donde T es la longitud del intervalo de señalización y E es la energía extendida durante este intervalo. La representación compleja pasa-baja es [12]:

$$s_{LP}(t) = \left(\frac{2E}{T}\right)^{\frac{1}{2}} e^{-i\phi(t)} \quad \text{Ec. 3.7}$$

La notación para modulaciones de fase continua, de la cual es parte la modulación GMSK, se representa como en Ec. 3.8, tomado de [13]:

$$\phi(t) = 2\pi \sum_{i \leq n} a_i h q(t - iT) \quad nT \leq t \leq (n+1)T \quad \text{Ec. 3.8}$$

De la ecuación Ec. 3.8 se desprende que:

- a_i es la secuencia de datos a transmitir, donde cada i -ésimo dato es tomado de un alfabeto M-ario $\pm 1, \pm 3, \dots \pm(M-1)$. Que para el caso de Bluetooth $M = 2$; entonces solo se tiene ± 1 .
- h es el índice de modulación, para Bluetooth está entre 0.28 a 0.32.
- $q(t)$ es la respuesta suave de fase, la cual describe cómo evoluciona la fase subyacente $2\pi a_i h$ con el tiempo. Esta es cualquier función que antes de un tiempo t es cero y después de un tiempo t es $1/2$.

La función $q(t)$ se define en [13]:

$$q(t) = \int_{-\infty}^t g(\tau) d\tau \quad \text{Ec. 3.9}$$

La función $g(\tau)$ representa el cambio de frecuencia de manera instantánea, las ecuaciones anteriores representan a las modulaciones en fase, el pulso $g(\tau)$ define que tipo de esquema es usado, puede tratarse de un *Raised Cosine de longitud L (LRC)*, *Tamed Frequency Modulation (TFM)*, *Spectral Raised Cosine de longitud L (LSRC)*,

Gaussian Minimum Shift Keying (GMSK) o Rectangular Frequency Pulse de longitud L (LREC) definidas en [13], el pulso GMSK es:

$$g(t) = \frac{1}{2T} \left\{ Q \left[2\pi B_b \frac{t - \frac{T}{2}}{(\ln 2)^{\frac{1}{2}}} \right] - Q \left[2\pi B_b \frac{t + \frac{T}{2}}{(\ln 2)^{\frac{1}{2}}} \right] \right\} \quad \text{Ec. 3.10}$$

Donde B_b es el producto de ancho de banda por periodo de bit (BT) con un valor de 0.5. Y la función $Q(t)$ es definida como:

$$Q(t) = \int_t^{\infty} \frac{1}{(2\pi)^{1/2}} e^{-\frac{\tau^2}{2}} d\tau \quad \text{Ec. 3.11}$$

Una manera de visualizar la Ec. 3.6 es usando la identidad trigonométrica del coseno de la suma de 2 ángulos, de donde se obtiene:

$$s(t, \alpha_n) = I(t) \cos(2\pi f_o t) - Q(t) \sin(2\pi f_o t) \quad \text{Ec. 3.12}$$

Donde:

$$I(t) = \cos[\phi(t, \alpha_n)] \quad \text{Ec. 3.13}$$

$$Q(t) = \sin[\phi(t, \alpha_n)]$$

Donde $\phi(t, \alpha_n)$ es la fase modulada y es una redefinición de la ecuación Ec. 3.8:

$$\phi(t, \alpha_n) = \theta(t, \alpha_n) + \theta_n \quad \text{Ec. 3.14}$$

donde:

$$\theta(t, \alpha_n) = 2\pi h \sum_{i=n-L+1}^n a_i q(t - iT)$$

$$\theta_n = \pi h \sum_{-\infty}^{n-L} a_i$$

θ_n representa el estado de fase, el número de estados correlativos es finito e igual a M^{L-1} , y $\theta(t, \alpha_n)$ es la rama de la fase del Trellis. Para índices de modulación racionales, se puede definir términos de números enteros como en la Ec. 3.15 [13]:

$$h = \frac{2k}{p}, \quad \text{donde } k, p \in \{Z\} \quad \text{Ec. 3.15}$$

3.3.1.2 Arquitectura del modulador GMSK.

Los requerimientos que debe cumplir el modulador GMSK deben ser:

- R-Tx-1:** Tener un índice de modulación de $0.32 \pm 1\%$.
- R-Tx-2:** Cumplir con las características de densidad espectral de potencia con un producto de ancho de banda por periodo de bit de 0.5.
- R-Tx-3:** Permitir la implementación del modulador en tecnología digital.

Para cumplir los requerimientos antes expuestos se realizó la programación de 3 diferentes arquitecturas en MATLAB, la primera se realiza por medio de datos en formato NRZ, pasados por un shaping Gaussiano y al final por un modulador FM (VCO) como la reportada en [11], las segunda implementación realizada fue por medio de una señal NRZ pasada por un modulador BPSK y limitando el espectro de potencia por un filtro Gaussiano, esta se encuentra reportada en [6], y la última fue la realizada en el trabajo previo de modulación de fase digital [14], considerando la arquitectura propuesta en [13], la cual es una implementación basada en un vector correlativo, y tablas de senos y cosenos.

Las 3 arquitecturas planteadas cumplieron con los requerimientos establecidos, por lo que se realizó un cuadro comparativo para determinar cuál de los moduladores es típico dentro de la bibliografía especializada.

Artículo	Transmisor
[6]	Filtro Gaussiano + modulador FSK (I-Q)
[8]	Modulador I-Q
[11]	Datos NRZ + filtro Gaussiano + VCO
[15]	No especificado
[16]	CPFSK en banda base + Filtro pasa bajas Gaussiano (I-Q)
[17]	Datos + filtro Gaussiano + VCO
[18]	Datos + filtro Gaussiano + VCO
[19]	I-Q ROM

[9]	Modulador I-Q
-----	---------------

Tabla 3.2 Cuadro comparativo de moduladores GMSK en diferentes artículos.

De la Tabla 3.2 se determinó que el modulador típico es aquel que genera las señales GMSK en banda base, es decir, un modulador I-Q. Una arquitectura que permite obtener estas señales es presentada en la Figura 3.6, reportada en [13] y [14].

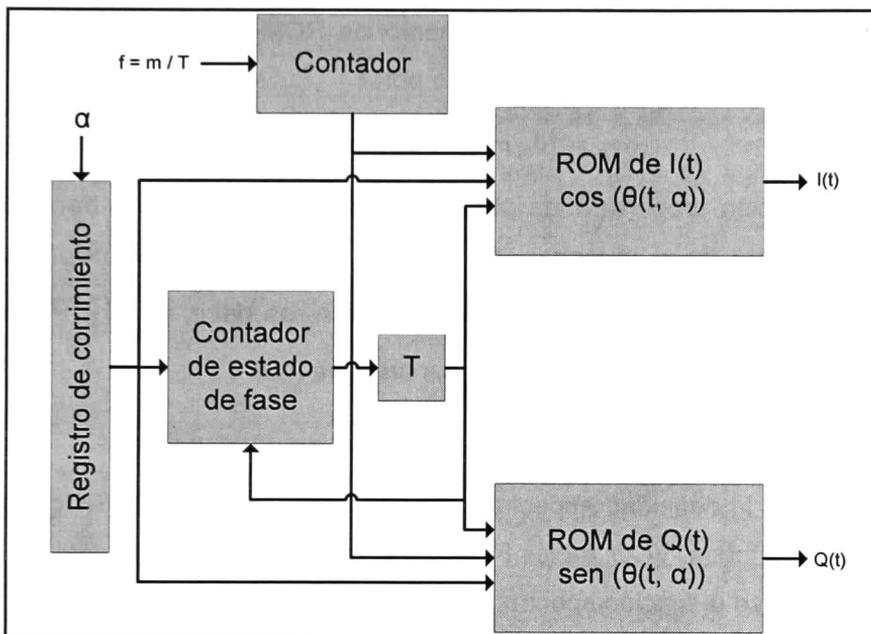


Figura 3.6 Implementación del Generador de la señal GMSK con ROM.

En la Figura 3.6 se muestra la arquitectura del modulador I-Q realizada en MATLAB y que servirá de modelo de referencia para la implementación en hardware del modulador GMSK. Esta arquitectura está constituida por un **registro de corrimiento** de longitud L , L representa la duración del pulso $g(t)$, su salida está representada por el vector de estado correlativo, en este vector se encuentran L datos binarios en formato NRZ, este vector se representa por la ecuación Ec. 3.16:

$$\alpha_n = [\alpha_{n-1}, \alpha_{n-2}, \dots, \alpha_{n-L+1}] \quad \text{Ec. 3.16}$$

La salida del **contador de estado de fase** es un valor del conjunto:

$$\theta_n \in \left\{ 0, \frac{2\pi}{p}, \frac{4\pi}{p}, \dots, \frac{(p-1)2\pi}{p} \right\} \quad \text{Ec. 3.17}$$

En cada intervalo de símbolo el valor siguiente del contador de estado de fase, definido como estado de fase siguiente θ_{n+1} , es calculado por el valor del vector correlativo α_n y el valor del estado de fase actual θ_n .

En las tablas **ROM** se guardan un conjunto de m muestras por símbolo a transmitir y para cada una de las ramas $I(t)$ y $Q(t)$, el intervalo de señalización por símbolo es T . El direccionamiento a las memorias ROM es por medio del valor del contador de estado del sistema representado por:

$$\varphi_n = [\theta_n, \alpha_{n-1}, \alpha_{n-2}, \dots, \alpha_{n-L+1}] \quad \text{Ec. 3.18}$$

Con la dirección proporcionada por φ_n se conoce el símbolo a transmitir, como cada símbolo consta de m muestras, se requiere de un **contador** que trabaja a una frecuencia de m veces la frecuencia de señalización, es decir, a $m \left(\frac{1}{T}\right)$. Con esto se accede a las m muestras por símbolo de las memorias ROM.

3.3.1.3 PSD de la señal GMSK generada por los moduladores.

Para calcular la densidad espectral de potencia de la señal GMSK generada por los 3 diferentes métodos: VCO, tablas ROM y modulación GFSK; se usó el algoritmo de Welch. En específico la función `spectrum.welch` de MATLAB.

Esta función consiste en usar un método de periodograma modificado y promediado de Welch, los pasos que sigue este algoritmo son los siguientes:

1. Segmentar el vector de entrada en 8 segmentos de igual longitud, con un traslape del 50%. La longitud del segmento es de 64.
2. Los segmentos son pasados por una ventana Hamming de la misma longitud del segmento.
3. Se aplica una `fft` a los datos ventaneados.
4. El periodograma de cada segmento ventaneado es calculado
5. Un conjunto de periodogramas son promediados para formar la estimación del espectro.
6. Se escala la estimación del espectro resultante para calcular la PSD.

En la Figura 3.7, se muestra la densidad espectral de potencia normalizada en frecuencia para los 3 moduladores implementados, se toma la densidad espectral de potencia de la señal MSK como marco comparativo, con esto se está cumpliendo con el requerimiento R-Tx-2:.

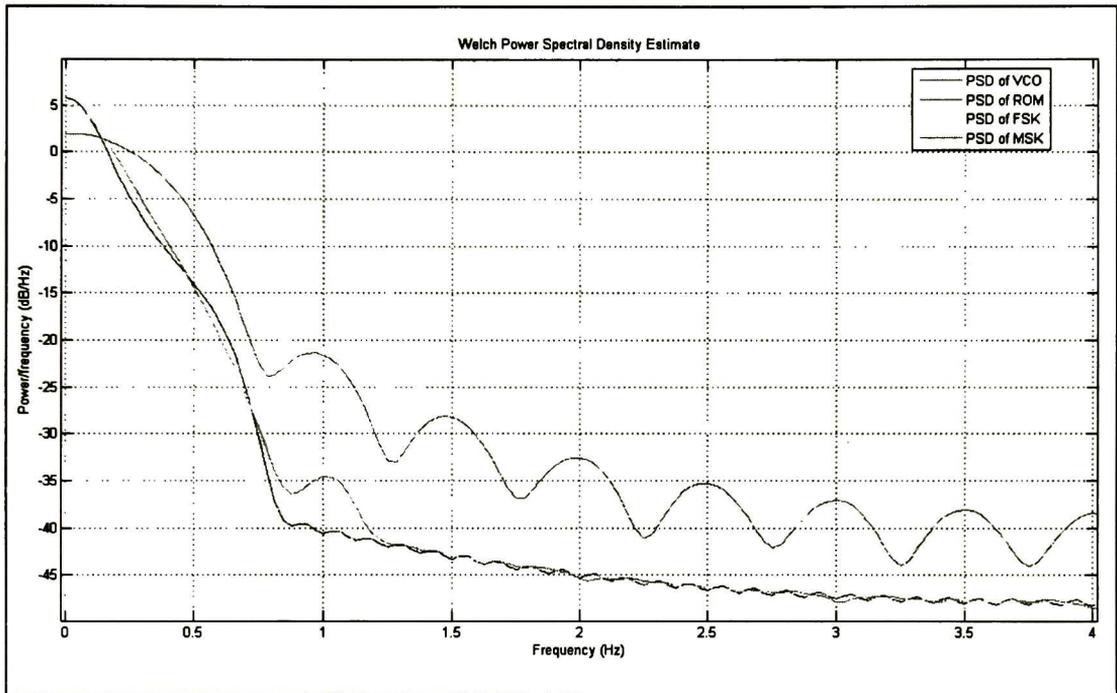


Figura 3.7 Densidad espectral de potencia normalizada para 3 moduladores GMSK.

3.3.2 Demodulador GMSK.

El demodulador fue el bloque que requirió más tiempo, ya que se encontró bastante documentación acerca de Bluetooth, pero en los reportes los receptores se presentaban como cajas negras, es decir, sin mostrar la arquitectura del demodulador implementada. Esto representa propiedad intelectual que los desarrolladores y/o investigadores guardan con recelo, en algunos reportes de bibliografía especializada donde se puede apreciar esto son: [6], [8], [19], [16], [9], [18].

Uno de los parámetros más importantes que debe cubrir el receptor Bluetooth es que debe cumplir un valor de 0.1% de BER, es decir, un valor de BER de 10^{-3} con una sensibilidad debajo ó igual a -70 dBm, en [9] realizando manipulaciones matemáticas

demuestran que el demodulador cumple este requerimiento con una relación señal a ruido de 21 dB, un dato importante a remarcar es que el estándar Bluetooth estuvo planteado inicialmente para hacer una demodulación analógica, y en este trabajo de tesis se pretende tener una arquitectura que permita implementarse a nivel digital.

Para ello se hace el planteamiento de los siguientes requerimientos que debe cumplir el demodulador GMSK:

R-RX-1: Cumplir para un valor de BER de 10^{-3} con una valor de 21 dB de relación señal a ruido.

R-RX-2: Tener una arquitectura del demodulador GMSK que permita implementarse en una plataforma de tecnología digital.

R-RX-3: Permitir la demodulación de señales GMSK generadas por moduladores I-Q y moduladores GMSK por medio de VCO ó GFSK.

Para cumplir con estos requerimientos se hace una exploración de los receptores GMSK, los cuales se pueden clasificar en 3 ramas principalmente:

- *Demoduladores no coherentes:* en este tipo de demodulación no se requiere del conocimiento de la señal portadora, pero el desempeño de la recepción se ve degradado.
- *Demoduladores coherentes:* en este tipo de recepción se requiere de circuitería extra para la recuperación de la señal portadora, pero esto mejora el desempeño del receptor.
- *Demoduladores mixtos:* en este caso se usa una portadora para bajar la señal a una frecuencia intermedia y de ahí se usa una demodulación no coherente para extraer la información de la señal GMSK [15].

Entre los demoduladores no coherentes para señales GMSK reportados en la bibliografía especializada se encuentra: el receptor de FM con detector de cruce por ceros y sincronía de símbolo [20]. En [15] se ven curvas comparativas del desempeño de BER vs E_b/N_0 para diferentes métodos de demodulación no coherentes: detector basado en PLL [21], discriminador de frecuencia, discriminador de línea de retraso y diferencial; de todos estos el que mejor desempeño tienes es el detector de cruce por ceros. Se realizó la implementación y simulación del detector de cruces por cero con un

filtro de integrate&dump, esta arquitectura permite recuperar la información de señales GMSK generadas con VCO's, cubriendo el valor de 0.1% de BER definido en la especificación, pero la demodulación para las señales GMSK generadas con moduladores I-Q, ya sea con tablas ROM ó GFSK, tiene un desempeño no aceptable, ya que presenta un floor-error, esto se visualiza en la Figura 3.8.

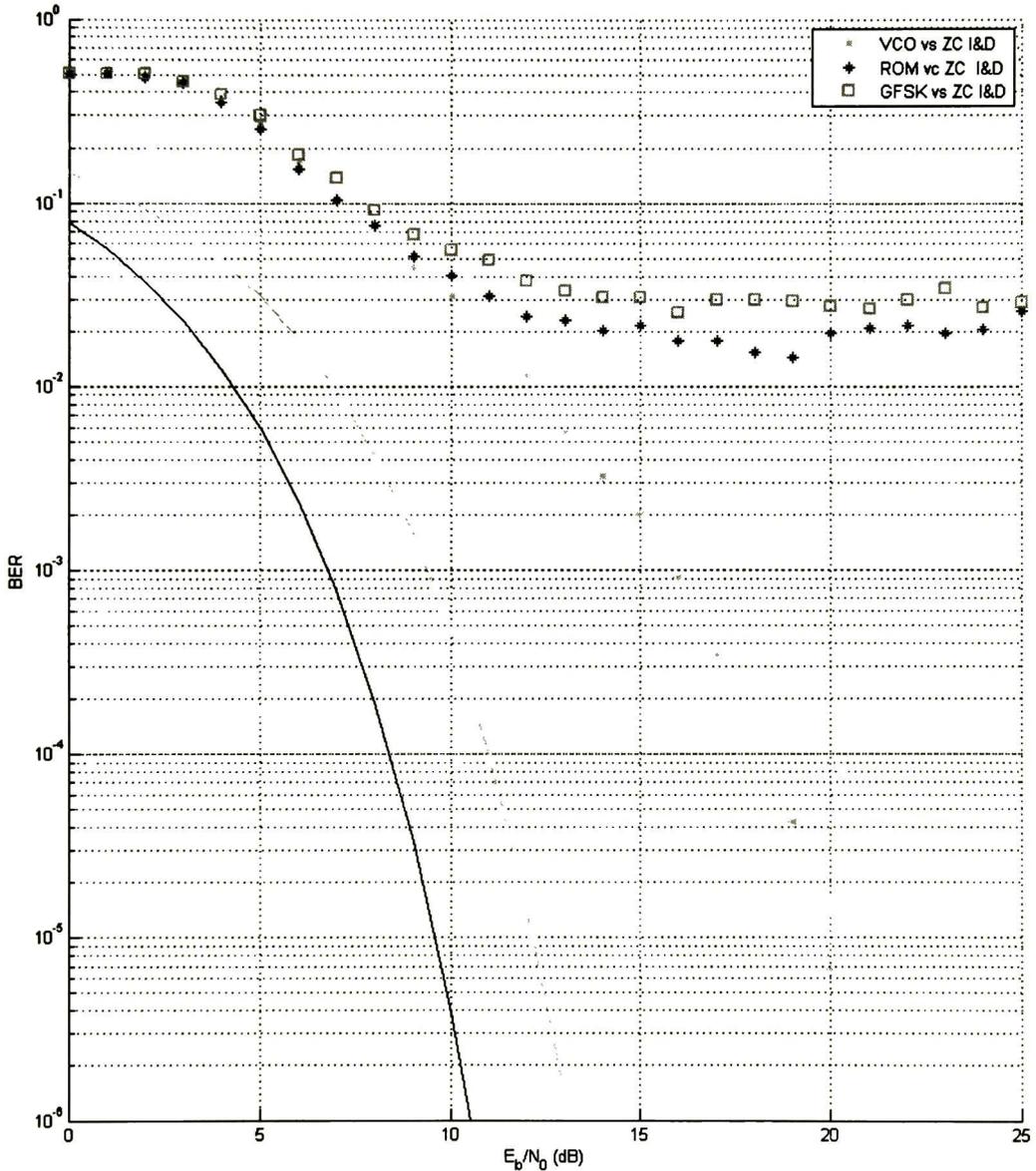


Figura 3.8 Desempeño BER vs EbNo del demodulador de cruce por ceros I&D para diferentes moduladores.

El primer demodulador coherente considerado, fue el demodulador óptimo, este emplea al algoritmo de Viterbi [13] y el árbol de Trellis para hacer la demodulación de la señal GMSK, el inconveniente de este demodulador es que requiere gran cantidad de memoria para guardar las diferentes ramas debido al desconocimiento del índice de modulación[10]. Este demodulador se encontró en el trabajo [14], al hacer simulaciones se requería el conocimiento exacto de los parámetros h ó p y L . Otro de los demoduladores coherentes considerado e implementado es el demodulador reportado en [22] denominado cuadracorrelacionador, la demodulación se lleva a cabo por medio de 2 derivadores, 2 multiplicadores y un sumador, el inconveniente se encontró al obtener las curvas de BER por simulación en MATLAB las cuales tenían un desempeño que no cubrían el estándar Bluetooth V1.2; también se probó el demodulador denominado "delay&multiply"[10], de igual manera no cubría con el estándar Bluetooth. El desempeño de estos dos demoduladores se puede apreciar en la Figura 3.9.

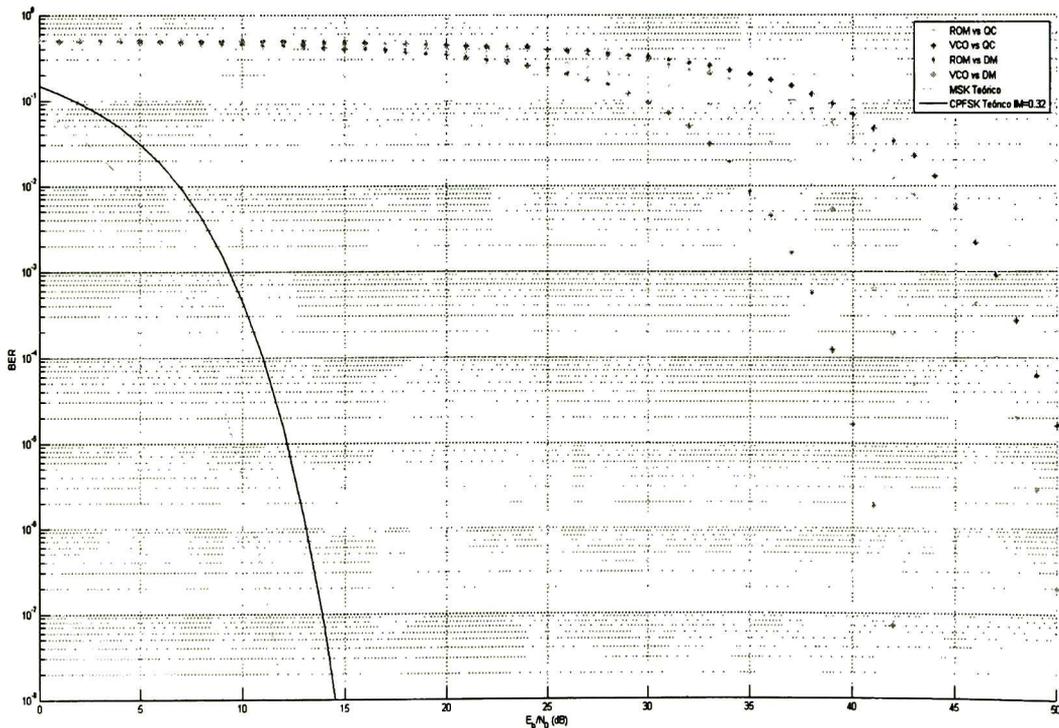


Figura 3.9 Desempeño BER vs E_b/N_0 de los demoduladores Delay&Multiply y Cuadracorrelator en banda base contra diferentes moduladores.

En [23], realizaron la implementación en DSP de un demodulador CPFSK para señal una señal con índice de modulación de 0.2, sin especificar que pulso de frecuencia instantáneo $q(t)$ usaron. Estos autores se basaron en [24] donde se presenta una arquitectura para demodulación CPFSK por medio de un bloque de estimación de máxima semejanza de N símbolos de observación, donde para cuando $N \rightarrow \infty$ la curvas de desempeño del demodulador tienden a las curvas teóricas. Las expresiones matemáticas expresadas en este artículo están planteadas para $h = 0.5$ y $g(t) = 1/2$, que se puede ver como MSK de acuerdo con el documento. No se reporta el desempeño de las curvas de BER contra E_b/N_0 para la señal GMSK con índice de modulación de $h = 0.32$, por lo que se desarrollaron las expresiones matemáticas y se realizó la simulación para medir el desempeño del demodulador para este caso.

3.3.2.1 Arquitectura del receptor GMSK.

La señal recibida está descrita por la ecuación Ec. 3.12 más ruido aditivo Gaussiano en el mejor de los casos, esto se muestra en la Figura 3.10: La señal recibida se multiplica por un coseno y un seno de la misma frecuencia de portadora, las componentes del doble de la frecuencia de portadora se eliminan por medio de un filtro pasa bajas, obteniendo como resultado: la señal en fase y en cuadratura. Las salidas constituyen la señal GMSK en su representación compleja pasa bajas descritas por la ecuación Ec. 3.13.

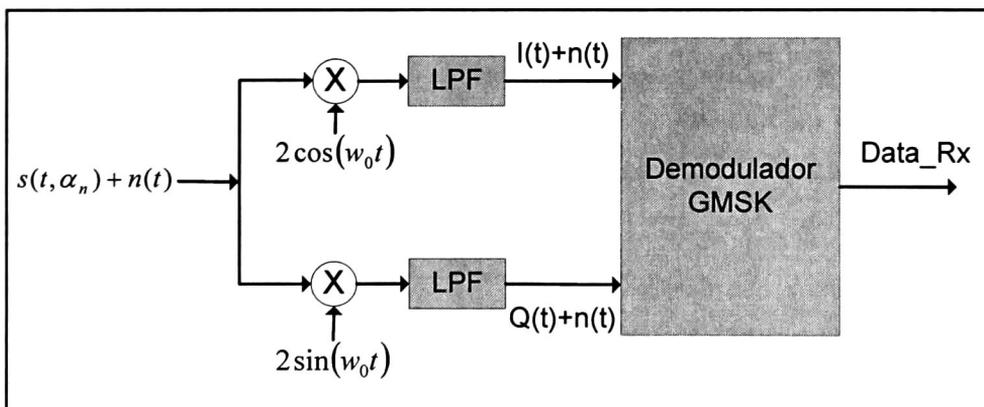


Figura 3.10 Demodulación coherente de la señal GMSK recibida.

La arquitectura del demodulador GMSK en banda base se muestra en la Figura 3.11, la arquitectura está constituida por **filtros** los cuales correlacionan con los 2 posibles valores de la señal enviada y un **bloque de estimación de máxima semejanza**, y al final un **comparador de magnitud**.

El diagrama de la Figura 3.11 se muestra la representación en tiempo continuo de la correlación y el proceso de recombicación, un equivalente en tiempo discreto es necesario para cumplir con el requerimiento **R-RX-2**. Las 4 multiplicaciones y las operaciones de integración se pueden reemplazar por 4 vectores de producto interno como en la **Error! Reference source not found.**

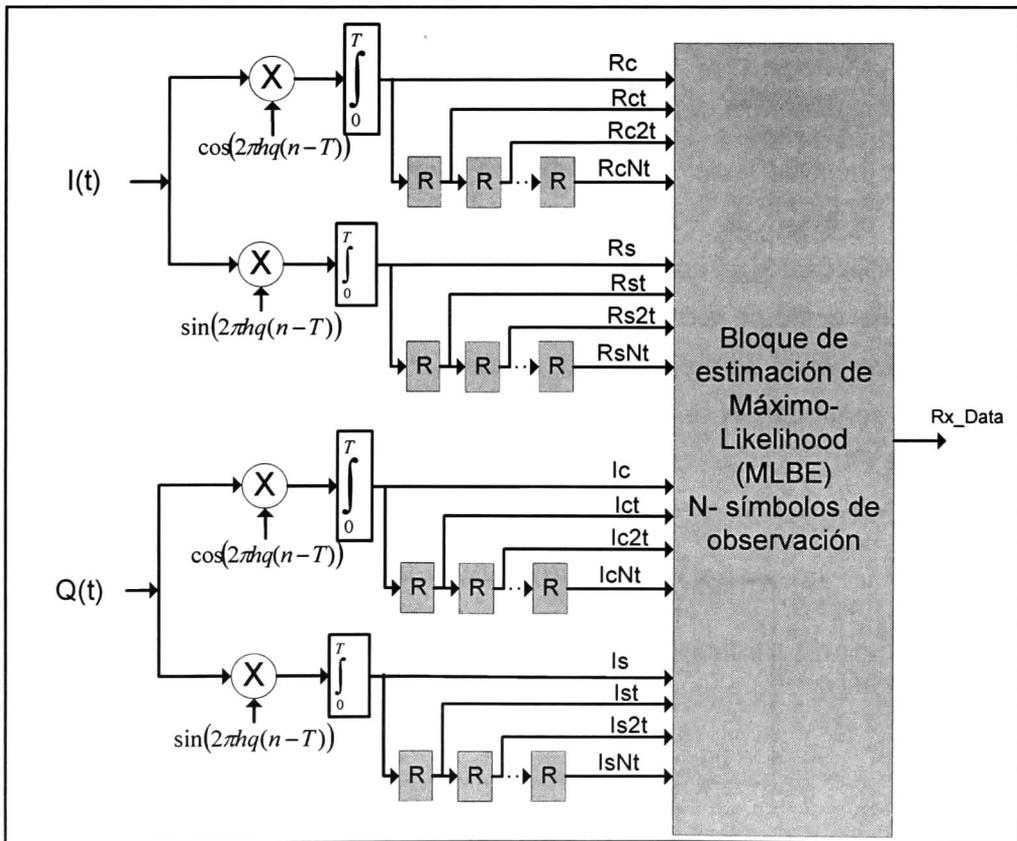


Figura 3.11 Arquitectura de demodulador GMSK en banda base.

$$R_c(kN) = I_{kN}^T X_c$$

$$R_s(kN) = I_{kN}^T X_s$$

Ec. 3.19

$$I_c(kN) = Q_{kN}^T X_c$$

$$I_s(kN) = Q_{kN}^T X_s$$

De la **Error! Reference source not found.**, se desprende que:

$$I_{kN} = [I(kN - 0), I(kN - 1), \dots, I(kN - N + 1)]^T$$

$$Q_{kN} = [Q(kN - 0), Q(kN - 1), \dots, Q(kN - N + 1)]^T$$

$$X_c = \left[\cos\left(\pi h \frac{0}{N}\right), \cos\left(\pi h \frac{1}{N}\right), \dots, \cos\left(\pi h \frac{N-1}{N}\right) \right]^T \quad \text{Ec. 3.20}$$

$$X_s = \left[\sin\left(\pi h \frac{0}{N}\right), \sin\left(\pi h \frac{1}{N}\right), \dots, \sin\left(\pi h \frac{N-1}{N}\right) \right]^T$$

La ecuación **Error! Reference source not found.** constituye las señales en tiempo discreto de las señales seno y coseno, además de la representación compleja pasa baja de la señal GMSK. Y donde k representa el periodo de bit bajo consideración.

Las salidas de las correlaciones son las entradas de un bloque de estimación de máxima semejanza (**MLBE**), con una ventana de observación de N símbolos, y al final un comparador de magnitud, por medio del cual se estima cual fue el bit enviado.

3.3.2.2 MLBE.

El bloque de estimación de máxima semejanza con una ventana de observación de N símbolos se presenta en [24], donde se hace el desarrollo matemático para una señal del esquema CPM con un pulso de frecuencia normalizado $g(t) = \frac{1}{2}$ e índice de modulación de 0.5, este se considera MSK.

El desarrollo de las ecuaciones para un pulso de frecuencia normalizado Gaussiano e índice de modulación de 0,32 no es presentado, por lo que considerando las ecuaciones planteadas en [24], se realiza el desarrollo matemático para la demodulación de señales GMSK, algunas expresiones de [24] se transcriben, por lo que tenemos:

$$\beta_l = \sum_{n=1}^N \Gamma_{l_n, N-n} C_n \quad \text{Ec. 3.21}$$

donde:

$$\Gamma_{i,j} = \int_{(n-j)T}^{(n-j+1)T} r(t) e^{-j2\pi h \Delta_{i_k}} dt \quad \text{Ec. 3.22}$$

y C_n es una constante compleja definida recursivamente como sigue:

$$C_1 = 1; \quad C_{k+1} = C_k e^{-j\pi h \Delta_{i_k}} \quad k = 1, 2, \dots, N \quad \text{Ec. 3.23}$$

El valor N es el tamaño de los símbolos observados y Δ_{i_k} es el vector de entrada de datos.

Como caso muestra se desarrollarán las ecuaciones para cuando $N = 3$, es decir, cuando se hace una observación de 3 bits para tomar una decisión del bit enviado.

$$\beta_{11} = \Gamma_{1,2} + e^{-j\pi h} \Gamma_{1,1} + e^{-j\pi h} e^{-j\pi h} \Gamma_{1,0} \quad \text{Ec. 3.24}$$

$$\beta_{12} = \Gamma_{1,2} + e^{-j\pi h} \Gamma_{1,1} + e^{-j\pi h} e^{-j\pi h} \Gamma_{2,0} \quad \text{Ec. 3.25}$$

$$\beta_{13} = \Gamma_{1,2} + e^{-j\pi h} \Gamma_{2,1} + e^{-j\pi h} e^{j\pi h} \Gamma_{1,0} \quad \text{Ec. 3.26}$$

$$\beta_{14} = \Gamma_{1,2} + e^{-j\pi h} \Gamma_{2,1} + e^{-j\pi h} e^{j\pi h} \Gamma_{2,0} \quad \text{Ec. 3.27}$$

$$\beta_{21} = \Gamma_{2,2} + e^{j\pi h} \Gamma_{1,1} + e^{j\pi h} e^{-j\pi h} \Gamma_{1,0} \quad \text{Ec. 3.28}$$

$$\beta_{22} = \Gamma_{2,2} + e^{j\pi h} \Gamma_{1,1} + e^{j\pi h} e^{-j\pi h} \Gamma_{2,0} \quad \text{Ec. 3.29}$$

$$\beta_{23} = \Gamma_{2,2} + e^{j\pi h} \Gamma_{2,1} + e^{j\pi h} e^{j\pi h} \Gamma_{1,0} \quad \text{Ec. 3.30}$$

$$\beta_{24} = \Gamma_{2,2} + e^{j\pi h} \Gamma_{2,1} + e^{j\pi h} e^{j\pi h} \Gamma_{2,0} \quad \text{Ec. 3.31}$$

Desarrollando la β_{11} tenemos:

$$\begin{aligned} \beta_{11} = & \int_{-2T}^{-T} r(t) e^{-w_m(t+2T)} dt + e^{-j\pi h} \int_{-T}^0 r(t) e^{-w_m(t+T)} dt \\ & + e^{-j\pi h} e^{-j\pi h} \int_0^T r(t) e^{-w_m(t)} dt \end{aligned} \quad \text{Ec. 3.32}$$

Donde $e^{-w_m(t)}$:

$$e^{-jw_m(t)} = \cos(w_m(t)) - j \sin(w_m(t)) \quad \text{Ec. 3.33}$$

Donde w_m

$$w_m = 2\pi h q(t - (n - j)T) \quad \text{Ec. 3.34}$$

Continuando el desarrollo de β_{11} :

$$\begin{aligned} \beta_{11} = & \int_{-2T}^{-T} (\text{Re}\{r(t)\} + j\text{Im}\{r(t)\}) [\cos(w_m(t + 2T)) - j \sin(w_m(t + 2T))] dt \\ & + e^{-j\pi h} \int_{-T}^0 (\text{Re}\{r(t)\} + j\text{Im}\{r(t)\}) [\cos(w_m(t + T)) - j \sin(w_m(t + T))] dt \quad \text{Ec. 3.35} \\ & + e^{-j\pi h} e^{-j\pi h} \int_0^T (\text{Re}\{r(t)\} + j\text{Im}\{r(t)\}) [\cos(w_m(t)) - j \sin(w_m(t))] dt \end{aligned}$$

Si se considera a $e^{-j\pi h}$, como:

$$e^{-j(\pi h)} = \cos(\pi h) - j \sin(\pi h) = f_1 - j f_2 \quad \text{Ec. 3.36}$$

Y si definimos las siguientes ecuaciones tenemos una reducción en las ecuaciones de la Ec. 3.49 - Ec. 3.62.

$$R_s = \int_0^T (\text{Re}\{r(t)\}) \sin(w_m t) dt \quad \text{Ec. 3.37}$$

$$R_c = \int_0^T (\text{Re}\{r(t)\}) \cos(w_m t) dt \quad \text{Ec. 3.38}$$

$$I_s = \int_0^T (\text{Im}\{r(t)\}) \sin(w_m t) dt \quad \text{Ec. 3.39}$$

$$I_c = \int_0^T (\text{Im}\{r(t)\}) \cos(w_m t) dt \quad \text{Ec. 3.40}$$

$$R_{sT} = \int_{-T}^0 (\text{Re}\{r(t)\}) \sin(w_m(t + T)) dt \quad \text{Ec. 3.41}$$

$$R_{cT} = \int_{-T}^0 (\text{Re}\{r(t)\}) \cos(w_m(t + T)) dt \quad \text{Ec. 3.42}$$

$$I_{sT} = \int_{-T}^0 (\text{Im}\{r(t)\}) \sin(w_m(t + T)) dt \quad \text{Ec. 3.43}$$

$$I_{cT} = \int_{-T}^0 (\text{Im}\{r(t)\}) \cos(w_m(t + T)) dt \quad \text{Ec. 3.44}$$

$$R_{s2T} = \int_{-2T}^{-T} (\text{Re}\{r(t)\}) \sin(w_m(t + 2T)) dt \quad \text{Ec. 3.45}$$

$$R_{c2T} = \int_{-2T}^{0-T} (\text{Re}\{r(t)\}) \cos(w_m(t + 2T)) dt \quad \text{Ec. 3.46}$$

$$I_{s2T} = \int_{-2T}^{-T} (\text{Im}\{r(t)\}) \sin(w_m(t + 2T)) dt \quad \text{Ec. 3.47}$$

$$I_{c2T} = \int_{-2T}^{-T} (\text{Im}\{r(t)\}) \cos(w_m(t + 2T)) dt \quad \text{Ec. 3.48}$$

Se realizó el desarrollo de las betas para diferentes tamaños de símbolos observados, esto es $N = 1, 2$ y 3 . Para cuando N es mayor a 3 , los desarrollos matemáticos se vuelven más complejos por lo que no son presentados. Solo se obtuvieron los resultados para $N > 3$ a través de simulación.

Ecuaciones para $N = 1$

$$\beta_{11} = (R_c + I_s) + j[-R_t + I_c] \quad \text{Ec. 3.49}$$

$$\beta_{21} = (R_c - I_s) + j[+R_t + I_c] \quad \text{Ec. 3.50}$$

Ecuaciones para $N = 2$

$$\beta_{11} = (R_{ct} + I_{st}) + f_1(R_c + I_s) + f_2(-R_s + I_c) - j[-(-R_{st} + I_{ct}) - f_1(-R_s + I_c) + f_2(R_c + I_s)] \quad \text{Ec. 3.51}$$

$$\beta_{12} = (R_{ct} - I_{st}) + f_1(R_c + I_s) - f_2(-R_s + I_c) - j[-(R_{st} + I_{ct}) - f_1(-R_s + I_c) - f_2(R_c + I_s)] \quad \text{Ec. 3.52}$$

$$\beta_{21} = (R_{ct} + I_{st}) + f_1(R_c - I_s) + f_2(R_s + I_c) - j[-(-R_{st} + I_{ct}) - f_1(R_s + I_c) + f_2(R_c - I_s)] \quad \text{Ec. 3.53}$$

$$\beta_{22} = (R_{ct} - I_{st}) + f_1(R_c - I_s) - f_2(R_s + I_c) - j[-(R_{st} + I_{ct}) - f_1(R_s + I_c) - f_2(R_c - I_s)] \quad \text{Ec. 3.54}$$

Ecuaciones para $N = 3$

$$\begin{aligned} \beta_{11} = & (R_{c2t} + I_{s2t}) + f_1(R_{ct} + I_{st}) + f_2(-R_{st} + I_{ct}) + (f_1^2 - f_2^2)(R_c + I_s) \\ & + 2f_2f_1(-R_s + I_c) \\ & - j[-(-R_{s2t} + I_{c2t}) - f_1(-R_{st} + I_{ct}) + f_2(R_{ct} + I_{st}) \\ & - (f_1^2 - f_2^2)(-R_s + I_c) + 2f_1f_2(R_c + I_s)] \end{aligned} \quad \text{Ec. 3.55}$$

$$\begin{aligned} \beta_{12} = & (R_{c2t} + I_{s2t}) + f_1(R_{ct} + I_{st}) + f_2(-R_{st} + I_{ct}) + (f_1^2 - f_2^2)(R_c - I_s) \\ & + 2f_2f_1(R_s + I_c) \\ & - j[-(-R_{s2t} + I_{c2t}) - f_1(-R_{st} + I_{ct}) + f_2(R_{ct} + I_{st}) \\ & - (f_1^2 - f_2^2)(R_s + I_c) + 2f_1f_2(R_c - I_s)] \end{aligned} \quad \text{Ec. 3.56}$$

$$\begin{aligned} \beta_{13} = & (R_{c2t} + I_{s2t}) + f_1(R_{ct} - I_{st}) + f_2(R_{st} + I_{ct}) + (R_c + I_s) \\ & - j[-(-R_{s2t} + I_{c2t}) - f_1(R_{st} + I_{ct}) + f_2(R_{ct} - I_{st}) - (-R_s + I_c)] \end{aligned} \quad \text{Ec. 3.57}$$

$$\begin{aligned} \beta_{14} = & (R_{c2t} + I_{s2t}) + f_1(R_{ct} - I_{st}) + f_2(R_{st} + I_{ct}) + (R_c - I_s) \\ & - j[-(-R_{s2t} + I_{c2t}) - f_1(R_{st} + I_{ct}) + f_2(R_{ct} - I_{st}) - (R_s + I_c)] \end{aligned} \quad \text{Ec. 3.58}$$

$$\begin{aligned} \beta_{21} = & (R_{c2t} - I_{s2t}) + f_1(R_{ct} + I_{st}) - f_2(-R_{st} + I_{ct}) + (R_c + I_s) \\ & - j[-(R_{s2t} + I_{c2t}) - f_1(-R_{st} + I_{ct}) - f_2(R_{ct} + I_{st}) - (-R_s + I_c)] \end{aligned} \quad \text{Ec. 3.59}$$

$$\begin{aligned} \beta_{22} = & (R_{c2t} - I_{s2t}) + f_1(R_{ct} + I_{st}) - f_2(-R_{st} + I_{ct}) + (R_c - I_s) \\ & - j[-(R_{s2t} + I_{c2t}) - f_1(-R_{st} + I_{ct}) - f_2(R_{ct} + I_{st}) - (R_s + I_c)] \end{aligned} \quad \text{Ec. 3.60}$$

$$\begin{aligned} \beta_{23} = & (R_{c2t} - I_{s2t}) + f_1(R_{ct} - I_{st}) - f_2(R_{st} + I_{ct}) + (f_1^2 - f_2^2)(R_c + I_s) \\ & - 2f_2f_1(-R_s + I_c) \\ & - j[-(R_{s2t} + I_{c2t}) - f_1(R_{st} + I_{ct}) - f_2(R_{ct} - I_{st}) \\ & - (f_1^2 - f_2^2)(-R_s + I_c) - 2f_1f_2(R_c + I_s)] \end{aligned} \quad \text{Ec. 3.61}$$

$$\begin{aligned} \beta_{24} = & (R_{c2t} - I_{s2t}) + f_1(R_{ct} - I_{st}) - f_2(R_{st} + I_{ct}) + (f_1^2 - f_2^2)(R_c - I_s) \\ & - 2f_2f_1(R_s + I_c) \\ & - j[-(R_{s2t} + I_{c2t}) - f_1(R_{st} + I_{ct}) - f_2(R_{ct} - I_{st}) - (f_1^2 - f_2^2)(R_s + I_c) \\ & - 2f_1f_2(R_c - I_s)] \end{aligned} \quad \text{Ec. 3.62}$$

3.3.2.3 Desempeño de BER del demodulador GMSK.

Para probar el desempeño del modulador se corrieron un conjunto de pruebas: se varió el parámetro N que define el número de símbolos observados en el demodulador para tomar una estimación del bit enviado, incrementando el parámetro N se incrementa la complejidad del receptor. En otra prueba se variaron los parámetros L y h en la construcción de la señal GMSK empleando la arquitectura propuesta en la sección 3.3.1.2. La última prueba consistió en demodular señales GMSK creadas por medio de VCO y un modulador GFSK (modulador BPSK y shaping Gaussiano) [6].

3.3.2.3.1 BER variando la cantidad N de símbolos observados.

Por medio de MATLAB se trabajó en un demodulador que permitiera realizar la recuperación de información de la señal GMSK producida por el modulador de tablas ROM. Se varió el parámetro N , es decir, la cantidad de símbolos observados para realizar la demodulación. El desempeño del demodulador comparando Bit Error Rate contra E_b/N_0 es el mostrado en la Figura 3.12. En este caso se dejó el índice de modulación fijo en 0.32 y la longitud de pulso Gaussiano en $L = 2$.

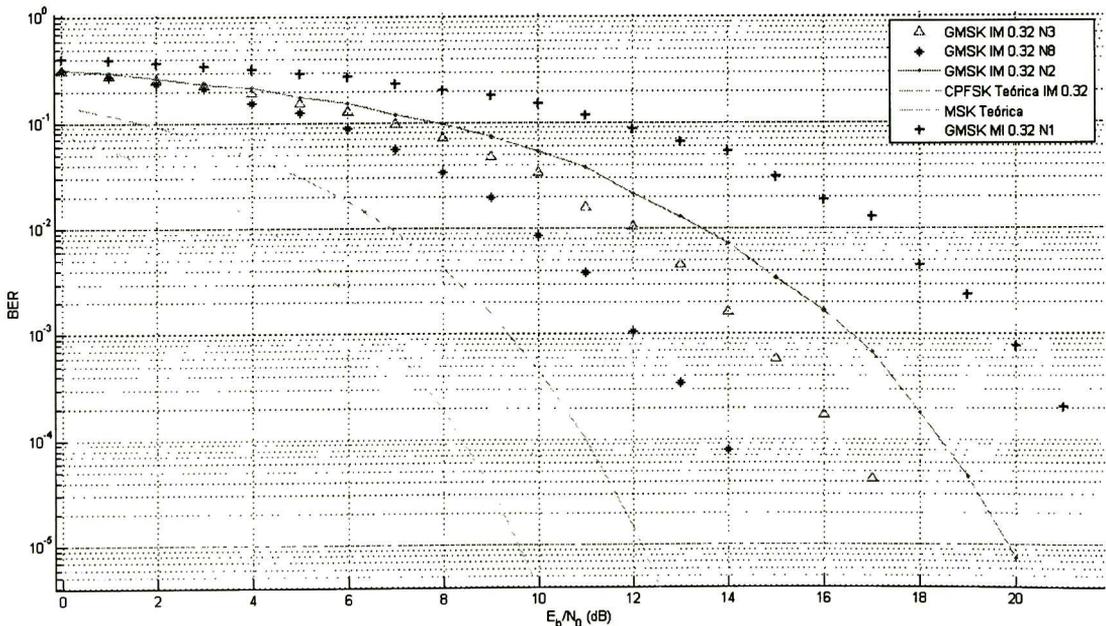


Figura 3.12 Desempeño del demodulador variando el número de símbolos observados N .

En la Tabla 3.3, se pueden comparar los diferentes desempeños de los demoduladores a un BER de 10^{-3} , el cual es un referente en el estándar Bluetooth. A manera de conclusión, a medida de que se incrementan los símbolos observados, es decir, cuando $N \rightarrow \infty$, la curva de BER tiende a la curva teórica. Del mismo modo, la complejidad del demodulador GMSK se incrementa a medida que N se aumenta. Por lo que existe un trade-off entre desempeño del modulador y su complejidad de implementación.

L, h	Símbolos observados para decisión			
	$N = 1$	$N = 2$	$N = 3$	$N = 8$
$L = 2, h = 0.32$	19.7 dB a 10^{-3}	16.5 dB a 10^{-3}	14.5 dB a 10^{-3}	12 dB a 10^{-3}

Tabla 3.3 Cuadro comparativo de los demoduladores para diferentes N 's y un BER a 10^{-3} .

3.3.2.3.2 BER variando la longitud L de la respuesta del pulso $g(t)$.

Como se demuestra en la Figura 3.13, el desempeño del demodulador propuesto es invariante a la longitud L de la respuesta del pulso Gaussiano. Para esta curva se consideró el índice de modulación fijo en 0.32 y la cantidad de bits observados se dejó fijo en $N = 3$ bits.

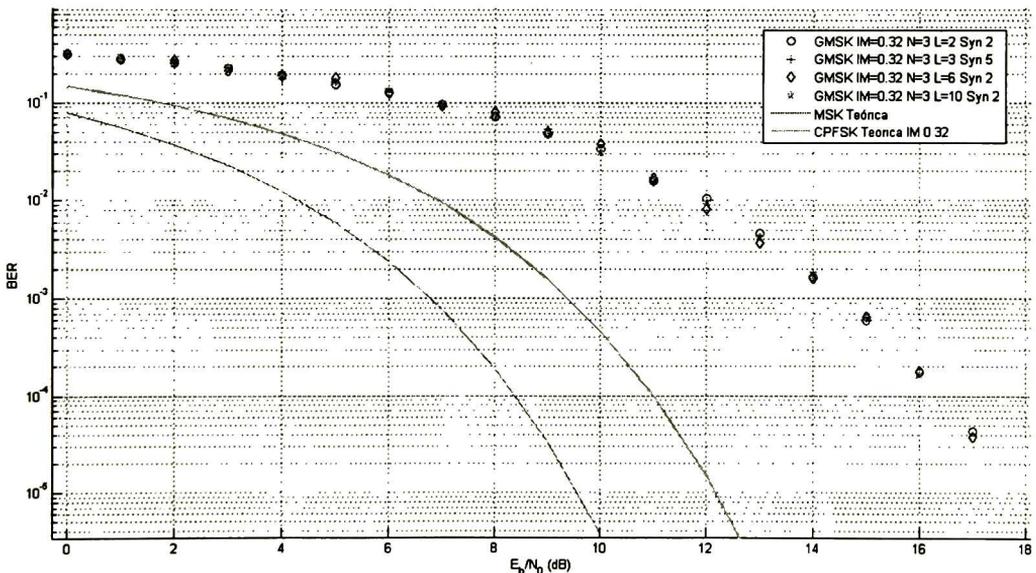


Figura 3.13 Desempeño del demodulador GMSK variando la longitud L del pulso Gaussiano y $N = 3$.

Dejando el parámetro $N = 2$, y el índice de modulación en $h = 0.32$, se varió el parámetro L en la construcción de la señal GMSK, obteniendo los resultados mostrados en la Figura 3.14, en esta grafica se muestra que el demodulador es robusto al cambio del parámetro L .

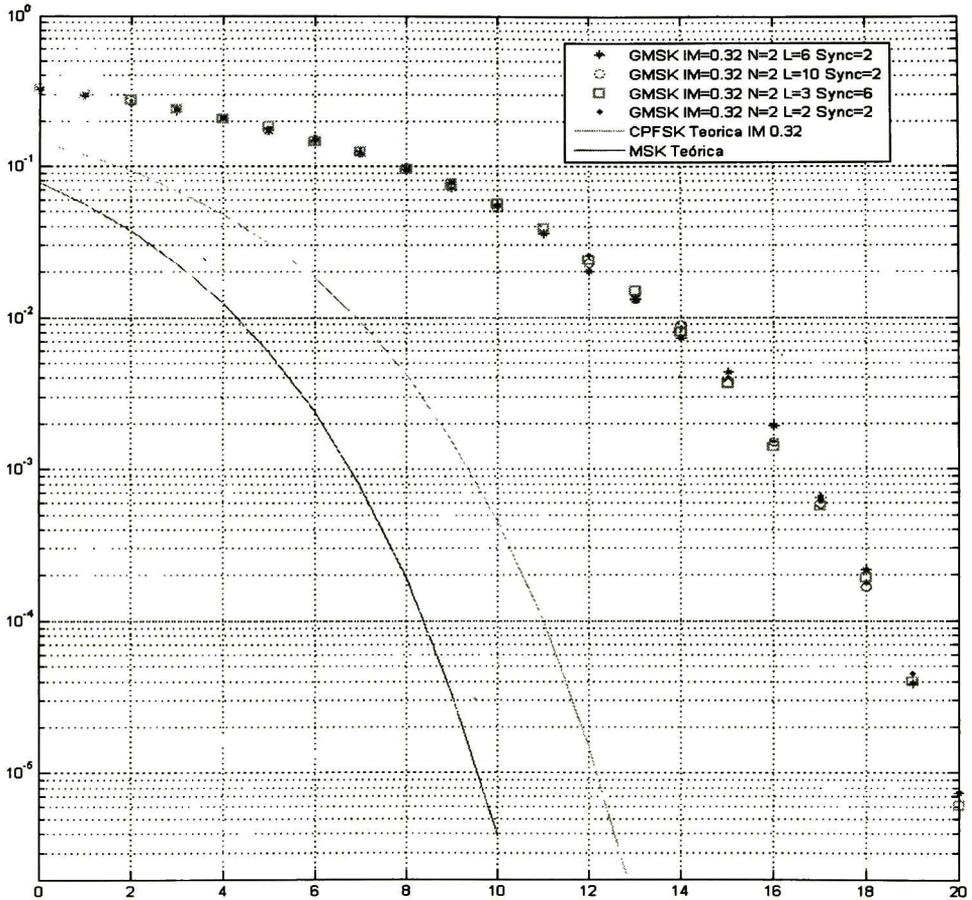


Figura 3.14 Desempeño del demodulador GMSK variando L del pulso Gaussiano y $N = 2$.

3.3.2.3.3 BER variando el índice de modulación h .

El estándar Bluetooth especifica que el índice de modulación debe estar entre 0.28 y 0.35, por lo que el demodulador debe de tener la capacidad de poder realizar la estimación de los bits enviados sin conocer el índice de modulación del transmisor. En la Figura 3.15, se muestra el desempeño del demodulador haciendo variar el índice de modulación en los casos límites de 0.28 y 0.35, también se muestra la curva típica con índice de modulación de 0.32, para este caso se consideró fijo $N = 3$, y $L = 2$.

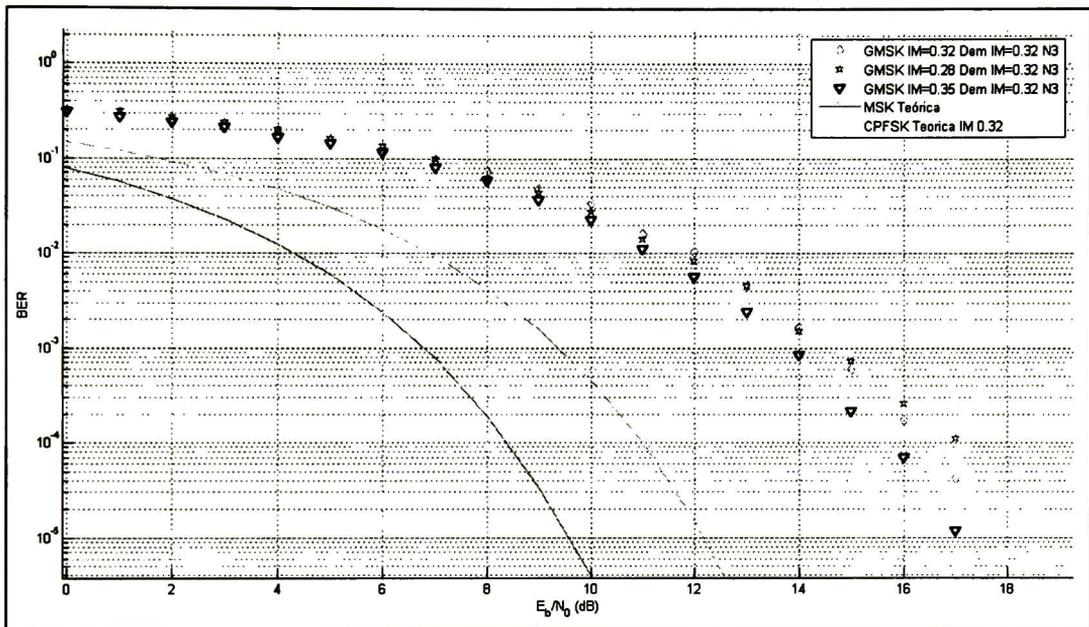


Figura 3.15 BER vs EbNo del demodulador GMSK, variando h y $N = 3$.

El desempeño del demodulador para $N = 2$ y $N = 3$ variando el índice de modulación es mostrado en el cuadro comparativo de la Tabla 3.4 a un valor de BER de 10^{-3}

Índice de modulación	Símbolos observados para decisión	
	$N = 2$	$N = 3$
0.35	15.7 dB a 10^{-3}	13.8 dB a 10^{-3}
0.32	16.5 dB a 10^{-3}	14.5 dB a 10^{-3}

0.28	17.4 dB a 10^{-3}	14.6 dB a 10^{-3}
------	---------------------	---------------------

Tabla 3.4 Cuadro comparativo para diferentes N 's e índices de modulación, comparados en 10^{-3}

Para el caso donde se varía el índice de modulación dejando los parámetros $N = 2$ y $L = 2$ fijos, es mostrado en la Figura 3.16.

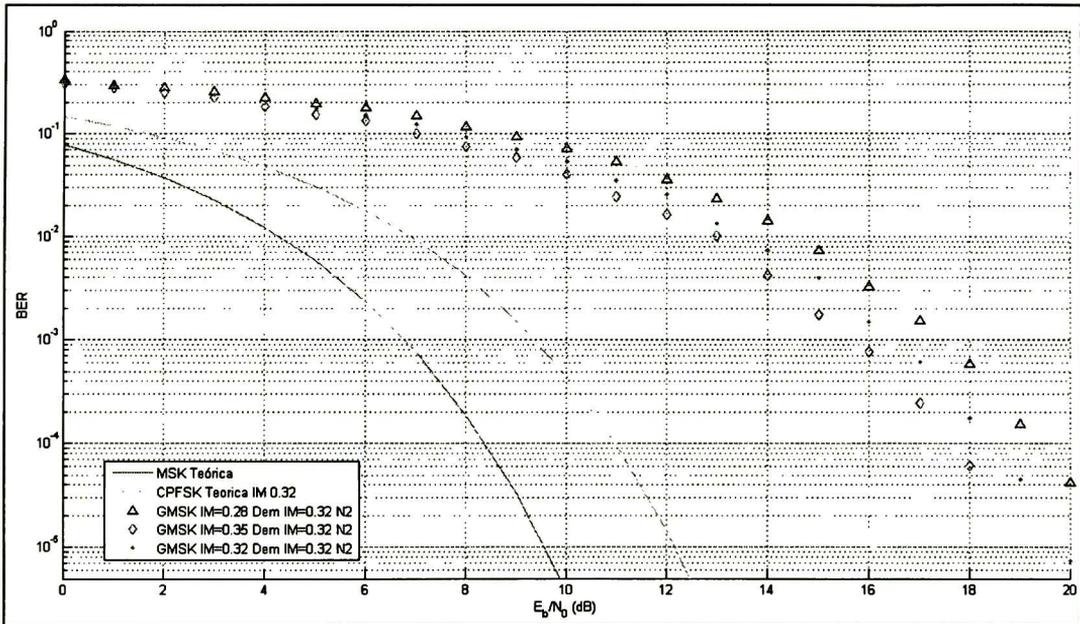


Figura 3.16 BER vs E_b/N_0 del demodulador GMSK, variando el índice de modulación h y $N = 2$.

El desempeño perdido en el demodulador no sobrepasa de 1 dB para cuando el índice de modulación vale 0.28. El demodulador sigue siendo robusto para cuando existen variaciones en el índice de modulación del transmisor.

3.3.2.3.4 BER variado técnica de modulación.

Se realizó la implementación de 3 moduladores GMSK: con tablas ROM's, VCO y GFSK (modulador BPSK y shaping Gaussiano). Las pruebas anteriores corresponden a la demodulación de señales creadas por medio de tablas ROM. La demodulación de señales GMSK por medio del modulador GFSK es mostrada en la Figura 3.17 Y el desempeño del demodulador para el caso cuando la señal GMSK es creada por medio de un VCO es mostrado en la Figura 3.18. Se concluye que el desempeño del

demodulador propuesto se mantiene para ambos casos, esto permite decir que el demodulador es robusto al tipo de generación de señal GMSK.

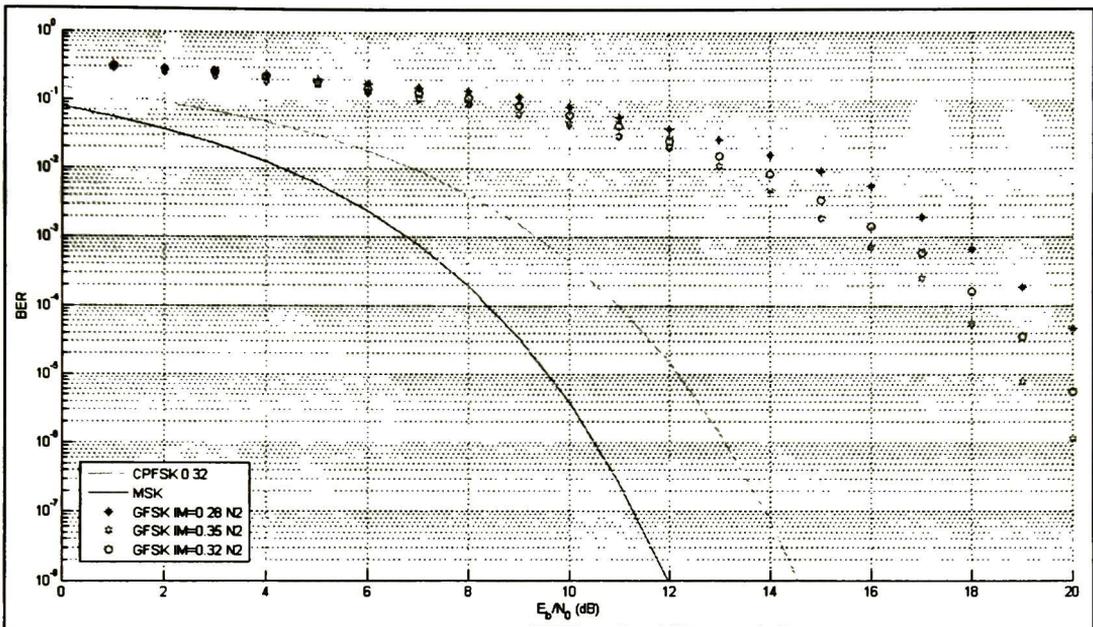


Figura 3.17 Curva de BER vs EbNo del desempeño del demodulador con modulación GFSK.

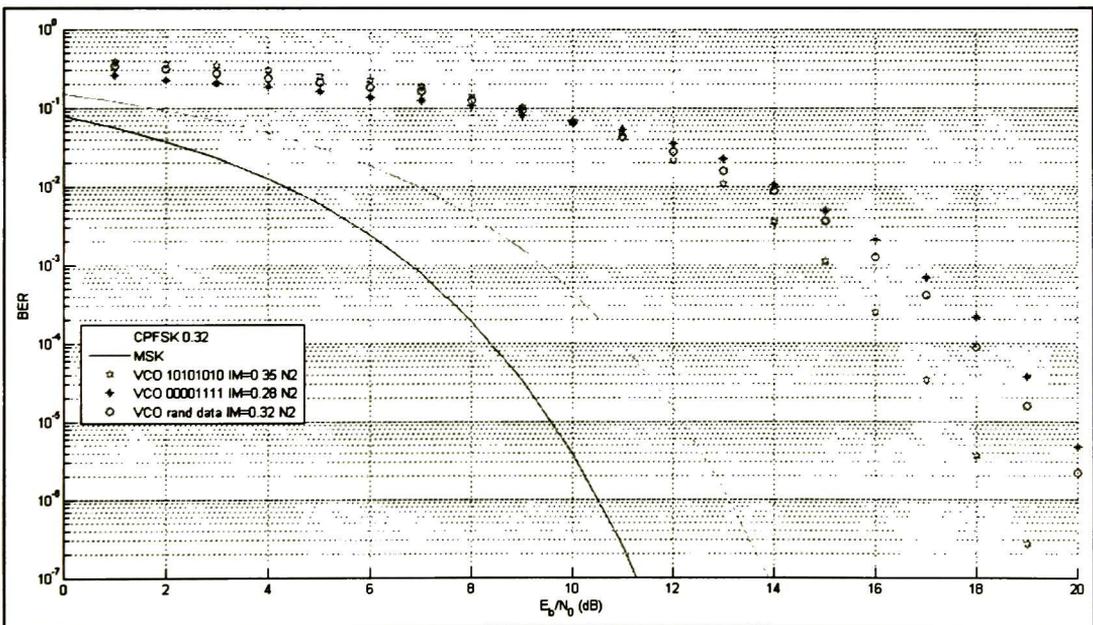


Figura 3.18 Desempeño del demodulador con modulación GMSK por medio de VCO.

3.3.2.4 Arquitectura del MLBE para $N = 3$ y $N = 2$.

Teniendo en cuenta las simulaciones anteriores y el desempeño alcanzado, se hizo una exploración de las arquitecturas para el bloque de estimación de máxima semejanza (MLBE) con $N = 2$ y $N = 3$ esto con el objetivo de revisar la complejidad de la futura implementación en hardware.

Para $N = 3$, se tiene un desempeño de BER a 10^{-3} de 14.5 dB en EbNo, la arquitectura se muestra en la Figura 3.19. En el caso de $N = 2$, el desempeño de BER a 10^{-3} es de 16.5 dB en EbNo, la arquitectura se muestra en la Figura 3.20.

El costo comparativo de cada una de las implementaciones en hardware se percibe en la Tabla 3.5, esto nos permite seleccionar la arquitectura de $N = 2$, por su costo beneficio entre complejidad y desempeño alcanzado, debido a que representa un poco menos de la mitad de los recursos comparados con la implementación en $N = 3$.

RECURSOS	Símbolos observados	
	$N = 2$	$N = 3$
Sumadores de 2 entradas	11	20
Sumadores de 3 entradas	8	0
Sumadores de 4 entradas	0	8
Sumadores de 5 entradas	0	8
Squares	8	16
Multiplicadores (corrimientos)	8	16
Comparadores de 2 entradas	3	7

Tabla 3.5 Cuadro comparativo entre implementaciones de $N = 2$ y $N = 3$ del MLBE

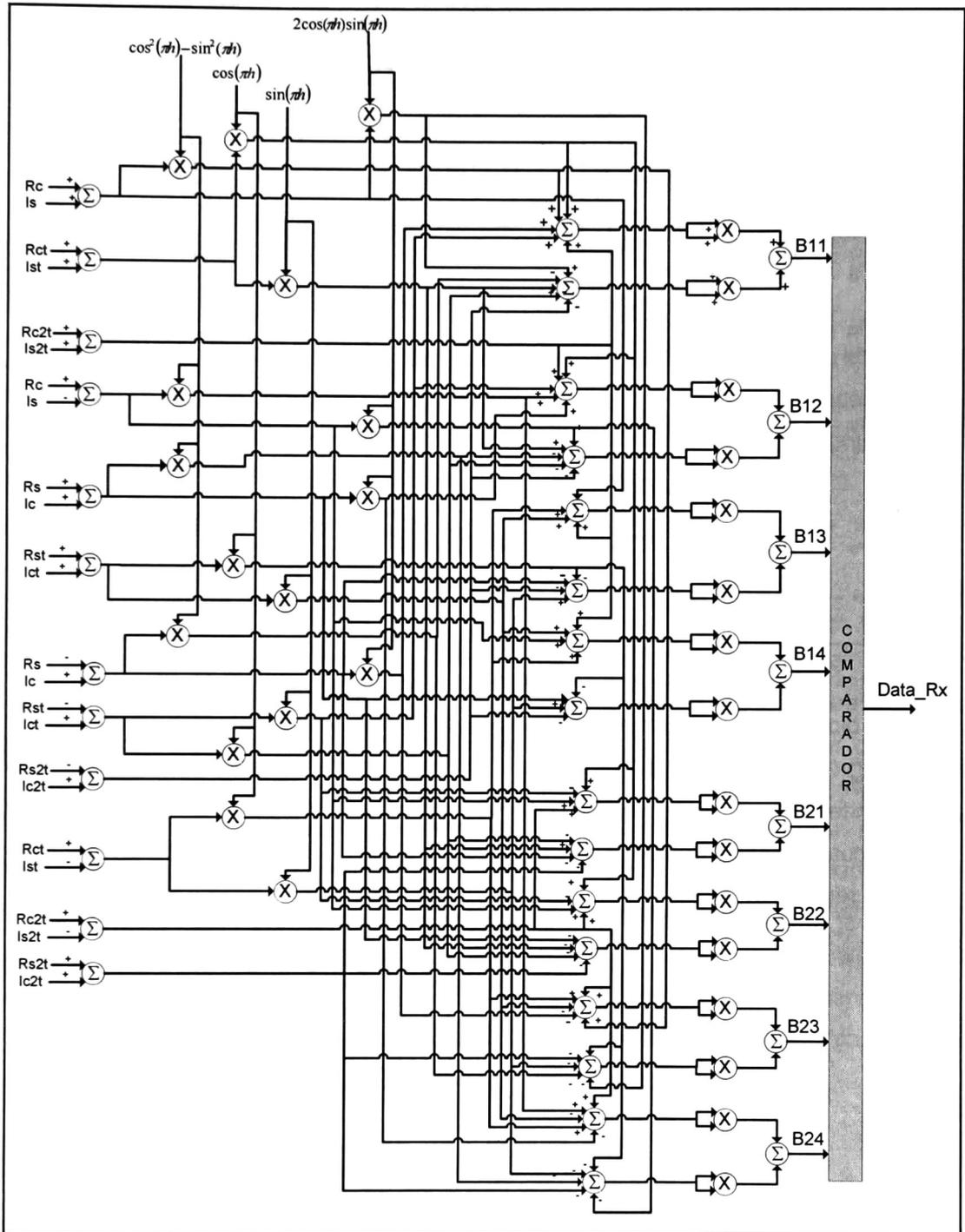


Figura 3.19 Arquitectura del bloque de estimación de máxima semejanza (MLBE) con $N = 3$.

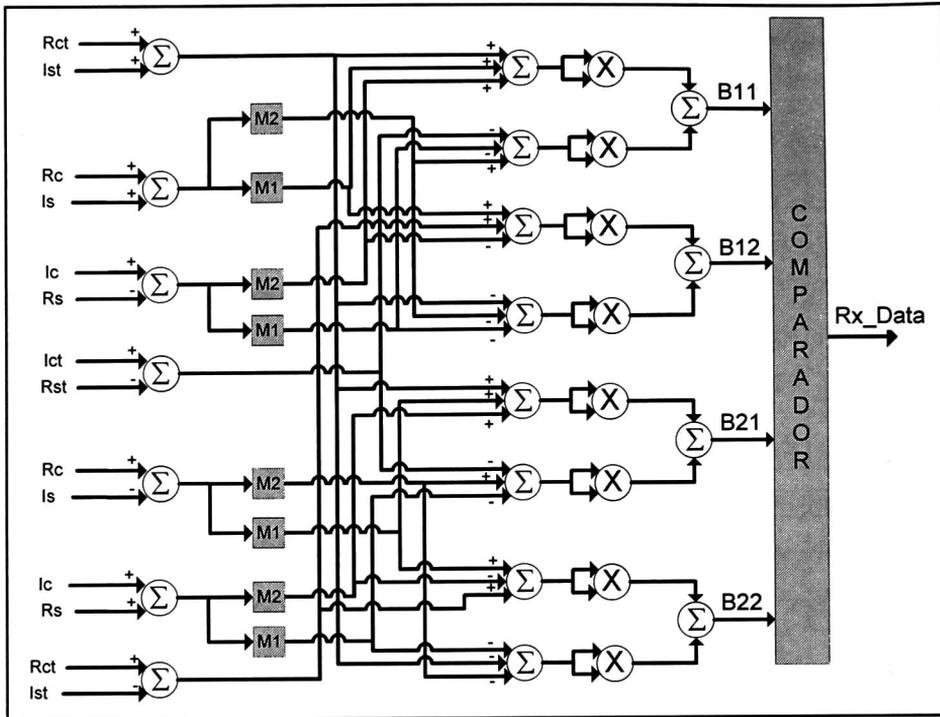


Figura 3.20 Arquitectura del bloque de estimación de máxima semejanza (MLBE) con $N = 2$.

Los bloques M1 y M2, son multiplicadores por constantes definidas como $\cos(\pi * h)$ y $\sin(\pi * h)$ en la arquitectura de $N = 2$, tomando el valor de $h = 0.31$ les corresponden los valores 0.56208 y 0.82708, respectivamente. Con aproximaciones de sumas y corrimientos como se muestra en la Figura 3.21, se obtiene un valor de 0.5625 y 0.8125, respectivamente.

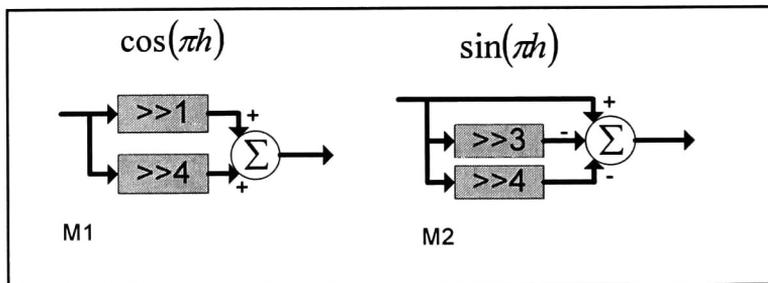


Figura 3.21 Aproximación de los multiplicadores por sumas y corrimientos.

CAPÍTULO 4 PRUEBAS Y RESULTADOS.

4.1 Introducción.

En este capítulo se presentan las pruebas ejecutadas a los módulos utilizados en la generación de las diferentes tramas de Bluetooth. Para esto se consideraron los datos muestra proporcionados por el estándar, las pruebas que requirieron más tiempo fueron precisamente aquellas orientadas al frequency hopping.

Así mismo se presenta el análisis en punto fijo o precisión finita para el caso del demodulador GMSK propuesto en la sección 3.3.2.1, esto para conocer los bits mínimos necesarios para que el desempeño del demodulador GMSK implementado conserve o se acerque al desempeño de la implementación en punto flotante.

4.2 Pruebas al generador de tramas Bluetooth.

4.2.1 Pruebas de caja blanca.

Las pruebas de caja blanca consistieron en la depuración del código para que cumpliera con las características de la trama, esta depuración se realizó para cada uno de los bloques que constituyen el procesador en banda base (generador de tramas Bluetooth). Los problemas en la programación del código de cada bloque se debieron principalmente a la interpretación de los valores descritos en el estándar, hay que hacer mención que el estándar especifica que usa un formato Little Endian para la transmisión de los bits, es decir, el primer dato en transmitirse al aire es el bit LSB, y la mayoría de los parámetros descritos en formato hexadecimal u octal estaban representados en formato Big Endian. El interpretar cuales valores estaban definidos con formato Little Endian ó cuales con Big Endian fue uno de los logros alcanzados al terminar las pruebas de caja blanca.

4.2.2 Pruebas de caja negra.

Para las pruebas de caja negra se consideraron los valores propuestos en el estándar Bluetooth V1.2 Vol. 2 Parte G [4]. Del estándar se tomaron los valores de entrada para los módulos y se hizo una comparación de los valores de salida. En la Tabla 4.1, se pueden observar las fechas del plan de validación de los bloques que constituyen el generador de tramas Bluetooth.

Nombre de TEST	Sección a Validar					Descripción	Modelo	Validación	Fecha
	Control de Timing y compensación de CLK	Access Code Generator	Preamble	Trailer	Sync Word Generator				
T-CLK-01	Control de Timing y compensación de CLK					Sumador	hecho	pasa	31/01/2011
T-AC-S1-LAP-01	Access Code Generator					42 LAP's diferentes	hecho	pasa	01/02/2011
	Preamble								
	Trailer								
	Sync Word Generator								
	Pseudo-random Noise Generation								
T-FH-S1-PSS-01	Page scan/Inquiry scan					CLKN= 0x00000000 UAP/LAP: 0x00000000 # ticks: cada 1000	hecho	pasa	02/02/2011
T-FH-S1-PI5-01	Page/Inquiry					CLKN= 0x00000000 UAP/LAP: 0x00000000 # ticks: cada 1	hecho	pasa	02/02/2011
T-FH-S1-SPR-01	Slave page response sequence					CLKN* = 0x00000010 UAP / LAP: 0x00000000	hecho	pasa	26/07/2011
T-FH-S1-MPR-01	Master page response sequence					Offset value: 24 CLKE* = 0x00000012 UAP / LAP: 0x00000000	hecho	pasa	27/07/2011
T-FH-S1-BCS-01	Basic channel hopping sequence					CLK start: 0x00000010 UAP/LAP: 0x00000000	hecho	pasa	04/02/2011
T-FH-S1-AFH-01	Adapted channel hopping sequence					CLK start: 0x00000010 ULAP: 0x00000000 Used Channels: 0x7fffffff	hecho	pasa	28/07/2011
T-FH-S1-AFH-02	Adapted channel hopping sequence					CLK start: 0x00000010 ULAP: 0x00000000 Used Channels: 0x7fffffff	hecho	pasa	28/07/2011
T-FH-S1-AFH-03	Adapted channel hopping sequence					CLK start: 0x00000010 ULAP: 0x00000000 Used Channels : 0x55555555	hecho	pasa	28/07/2011
T-FH-S1-AFH-04	Adapted channel hopping sequence					CLK start: 0x00000010 ULAP: 0x00000000 Used Channels : 0x2aaaaaaaaa	hecho	pasa	28/07/2011
T-FH-S2-PSS-01	Page scan/Inquiry scan					CLKN=0x00000000, UAP/LAP 0x2a96ef25 # ticks cada 1000	hecho	pasa	02/02/2011

T-FH-S2-PI5-01	Page/Inquiry	CLKN=0x00000000 UAP/LAP: 0x2a96ef25 #ticks: cada 1	hecho	pasa	02/02/2011
T-FH-S2-SPR-01	Slave page response sequence	CLKN* = 0x00000010 UAP / LAP: 0x2a96ef25	hecho	pasa	26/07/2011
T-FH-S2-MPR-01	Master page response sequence	Offset value: 24 CLKE* = 0x00000012 UAP / LAP: 0x2a96ef25	hecho	pasa	27/07/2011
T-FH-S2-BCS-01	Basic channel hopping sequence	CLK start: 0x00000010 UAP/LAP: 0x2a96ef25	hecho	pasa	04/02/2011
T-FH-S2-AFH-01	Adapted channel hopping sequence	CLK start: 0x00000010 U LAP: 0x2a96ef25 Used Channels:0x7fffffffffffffff	hecho	pasa	28/07/2011
T-FH-S2-AFH-02	Adapted channel hopping sequence	CLK start: 0x00000010 U LAP: 0x2a96ef25 Used Channels:0x7fffffffffc00000	hecho	pasa	28/07/2011
T-FH-S2-AFH-03	Adapted channel hopping sequence	CLK start: 0x00000010 U LAP: 0x2a96ef25 Used Channels : 0x5555555555555555	hecho	pasa	28/07/2011
T-FH-S2-AFH-04	Adapted channel hopping sequence	CLK start: 0x00000010 U LAP: 0x2a96ef25 Used Channels : 0x2aaaaaaaaaaaaaaaaa	hecho	pasa	28/07/2011
T-FH-S3-PSS-01	Page scan/Inquiry scan	CLKN=0x00000000, UAP/LAP: 0x6587cba9 # ticks: cada 1000	hecho	pasa	02/02/2011
T-FH-S3-PI5-01	Page/Inquiry	CLKN=0x00000000 UAP/LAP: 0x6587cba9 #ticks: cada 1	hecho	pasa	02/02/2011
T-FH-S3-SPR-01	Slave page response sequence	CLKN* = 0x00000010 UAP / LAP: 0x6587cba9	hecho	pasa	26/07/2011
T-FH-S3-MPR-01	Master page response sequence	Offset value: 24 CLKE* = 0x00000012 UAP / LAP: 0x6587cba9	hecho	pasa	27/07/2011
T-FH-S3-BCS-01	Basic channel hopping sequence	CLK start: 0x00000010 UAP/LAP: 0x6587cba9	hecho	pasa	04/02/2011
T-FH-S3-AFH-01	Adapted channel hopping sequence	CLK start: 0x00000010 U LAP: 0x6587cba9 Used Channels:0x7fffffffffffffff	hecho	pasa	28/07/2011

Capítulo 4 Pruebas y resultados.

T-FH-S3-AFH-02	Adapted channel hopping sequence	CLK start: 0x0000010 ULAP: 0x6587cba9 Used Channels: 0x7fffffffcc00000	hecho	pasa	28/07/2011
T-FH-S3-AFH-03	Adapted channel hopping sequence	CLK start: 0x0000010 ULAP: 0x6587cba9 Used Channels : 0x55555555555555555555	hecho	pasa	28/07/2011
T-FH-S3-AFH-04	Adapted channel hopping sequence	CLK start: 0x0000010 ULAP: 0x6587cba9 Used Channels : 0x2aaaaaaaaaaaaaaaaaaaa	hecho	pasa	28/07/2011
T-PH-HEC-01	HEC y packet header	20 UAP's diferentes	hecho	pasa	07/02/2011
T-CRC-TX-01	Generación de CRC	Generación de CRC de 10 bytes de información	hecho	pasa	08/02/2011
T-PK-DH1	Paquetes	Paquete DH1	hecho	pasa	10/02/2011
T-PK-DM1	Whitening	Paquete DM1	hecho	pasa	11/02/2011
T-WH-S1	FEC	Generación de la secuencia de Whitening	hecho	pasa	09/02/2011
T-FEC-01	FEC	FEC con rate 2/3	hecho	pasa	09/02/2011
T-FEC-02	Detección de CRC-16	FEC con rate 1/3	hecho	pasa	09/02/2011
T-CRC-RX-01	Detección de CRC-16	Detección del CRC en recepción	hecho	pasa	08/02/2011

Tabla 4.1 Fechas de validación de los bloques del generador de tramas Bluetooth.

4.3 Análisis en punto fijo del demodulador GMSK.

En esta sección se plantea un análisis de punto fijo para el demodulador GMSK propuesto, el script implementado en MATLAB del modulador GMSK propuesto se modificó por medio de la función "fi" de MATLAB para adecuarlo a precisión finita.

La función "fi" permite construir un objeto en punto fijo. Un ejemplo de su uso se define a continuación:

```
% a = fi(v, s, w, f)
a = fi(pi,1,15,12)
a =
    3.141601562500000

    DataTypeMode: Fixed-point: binary point scaling
    Signedness: Signed
    WordLength: 15
    FractionLength: 12
```

Figura 4.1 Uso de la función "fi" en MATLAB.

En la función "fi", se especifica el valor v que se desea establecer en precisión finita, el valor s especifica si el valor binario es no signado ($s = 0$) ó signado ($s = 1$), con w se especifica la cantidad de bits utilizados en total para representar la palabra binaria, incluyendo signo, fracción decimal y entera; y por último con f se especifica la cantidad de bits utilizados de parte fraccional.

Para el caso mostrado en la Figura 4.1, se realiza la aproximación de π en precisión finita, para ello se define el valor de $s = 1$, es decir, la palabra binaria está signada, el valor de $w = 15$ por lo que la longitud total de la palabra binaria es de 15 bits, y $f = 12$ que define la los bits que representan a la parte fraccional, por lo que de manera implícita se están definiendo 2 bits para la representación de la parte entera.

El demodulador GMSK propuesto consta de los bloques internos mostrados en la Figura 4.2, para el análisis de punto fijo se fue analizando la cantidad de bits necesarios para cada una de las etapas.

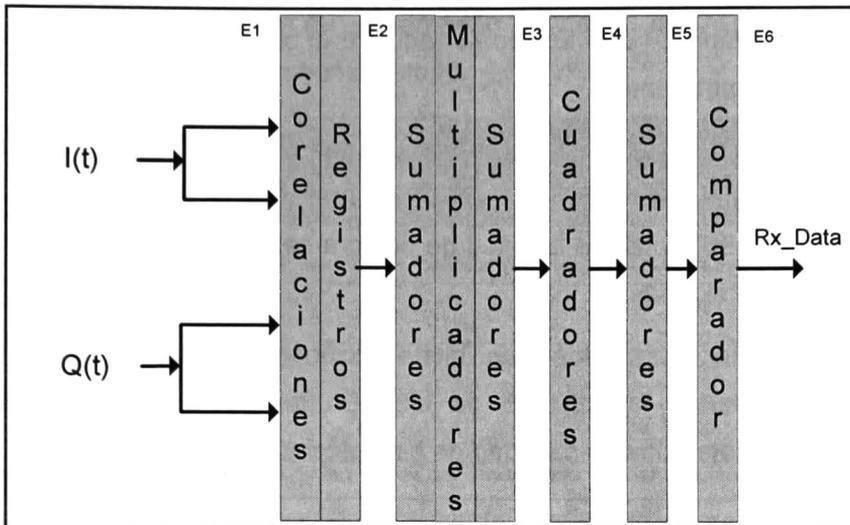


Figura 4.2 Bloques generales del demodulador GMSK.

En la medida que el valor de E_b/N_0 se incrementa, los valores máximos en cada una de las etapas van disminuyendo hasta quedar en un valor constante, esto se muestra en la Tabla 4.2.

Etapa/ E_b/N_0	0 dB	5 dB	10 dB	15 dB	20 dB
E1	± 11	± 6	± 4	± 3	± 2
E2	± 25	± 17	± 13	± 11	± 10
E3	± 35	± 34	± 25	± 21	± 19
E4	$+ 2.1 \times 10^{-3}$	+ 512	+ 511	+ 454	+ 374
E5	$+ 2.1 \times 10^{-3}$	+ 512	+ 511	+ 454	+ 374
E6	$+ 2.1 \times 10^{-3}$	+ 512	+ 511	+ 454	+ 374

Tabla 4.2 Valores máximos de cada etapa del demodulador GMSK.

De la tabla Tabla 4.2, se pueden hacer varias apreciaciones:

1. Si se define un rango de E_b/N_0 sobre el cual trabajará el demodulador GMSK, por decir, de 10 dB a 20 dB, los bits requeridos para la etapa E1 (entrada del demodulador) serían: 1 bit de signo, y 3 bits de parte entera. En la etapa E3, se requiere 1 bit de signo y 5 bits de parte entera.

2. De la etapa E4 a la E6, no se requiere el bit de signo, y se requieren de 9 bits de parte entera.

Para tener una longitud de palabra homogénea en los bloques del demodulador se propone utilizar para las etapas E1 E3 un bit de signo y 5 bits de parte entera, para las etapas de E4 – E6, donde la longitud de palabra se ve casi duplicada se proponen usar 9 bits sin bit de signo.

Para determinar la cantidad de bits fraccionarios necesarios, se realizaron simulaciones recortando la cantidad de bits. En la Figura 4.3 se muestran las simulaciones para determinar la cantidad de bits fraccionarios.

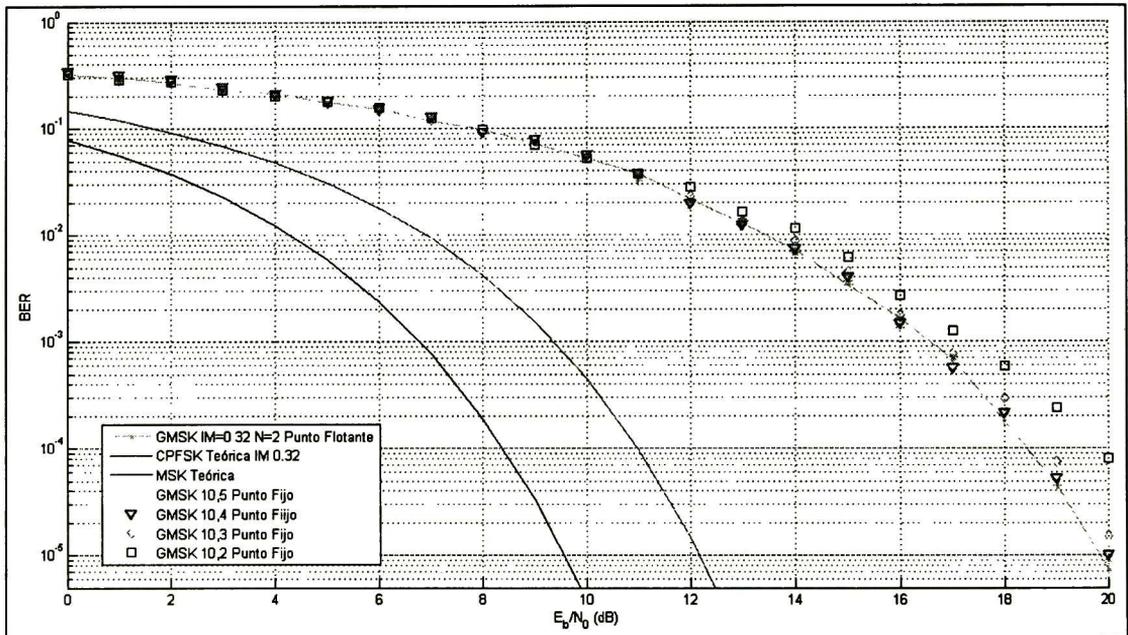


Figura 4.3 BER vs EbNo comparativo para determinar la cantidad de bits fraccionarios.

Para llevar a cabo las simulaciones de la Figura 4.3, en el script de MATLAB se dejó fija la cantidad de bits enteros a 10, y por cada nueva simulación se disminuyó en 1 la cantidad de bits fraccionarios, empezando con 5 bits y terminando con 2 bits, en este último el desempeño cae más de 1 dB.

De la Tabla 4.3 se puede afirmar que los bits fraccionarios necesarios para representar el demodulador GMSK son sólo cuatro. Ya que se tiene una pérdida muy

pequeña de apenas 0.03 dB de desempeño de BER contra EbNo respecto al desempeño del demodulador en punto flotante.

Bits fraccionarios	Perdida en dB respecto a la desempeño en punto fijo en 10^{-4} .
5	Despreciable
4	0.08 dB
3	0.33 dB
2	1.3 dB

Tabla 4.3 Tabla comparativa de de la perdida del desempeño del demodulador en punto fijo.

En la Figura 4.4, se puede observar como quedaron distribuidas las longitudes de palabra binaria para los bloques del demodulador GMSK propuesto.

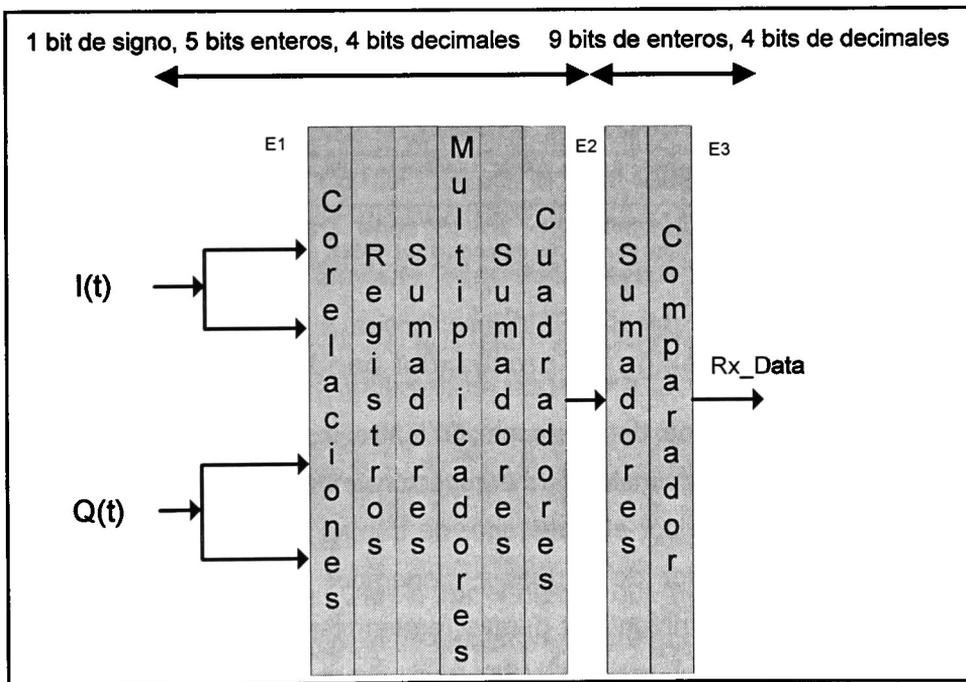


Figura 4.4 Bloques del demodulador GMSK en punto fijo.

4.3.1 Curvas de BER del demodulador GMSK en punto fijo.

Con las longitudes de palabra propuestas, el desempeño del demodulador GMSK se mantiene como se muestra en la Figura 4.5.

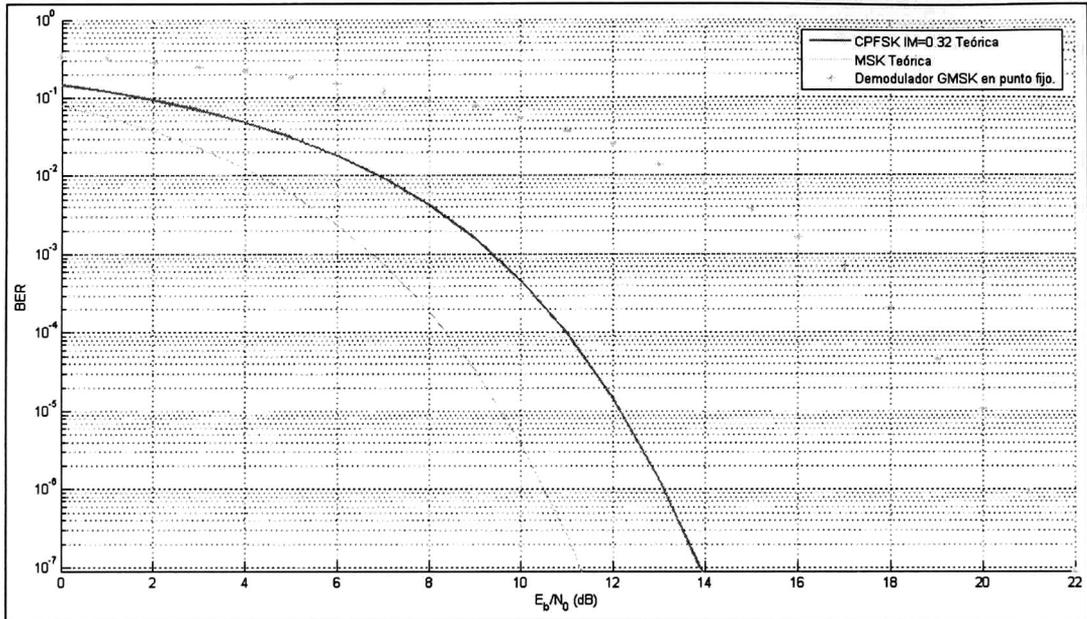


Figura 4.5 Desempeño del demodulador GMSK en punto fijo.

4.4 Pruebas de interconexión del procesador banda base.

Para hacer las pruebas de interconexión entre los bloques que conforman el estándar Bluetooth V1.2, se consideró la arquitectura mostrada en la Figura 4.6, estos bloques están conformados por el Generador de tramas de Bluetooth, constituido por los bloques de ensamblador de paquetes, generador de CRC's, whitening, FEC, generador de palabra de sincronía, y demás requerimientos de la generación de los paquetes Bluetooth; el modulador GMSK; un bloque que adiciona ruido aditivo Gaussiano a la forma de onda de la representación compleja pasa-baja de la señal GMSK; realizando la demodulación por medio del demodulador GMSK propuesto y la recuperación de la información útil es realizada por el decodificador de paquetes, al final se hace un comparativo entre datos enviados y recibidos.

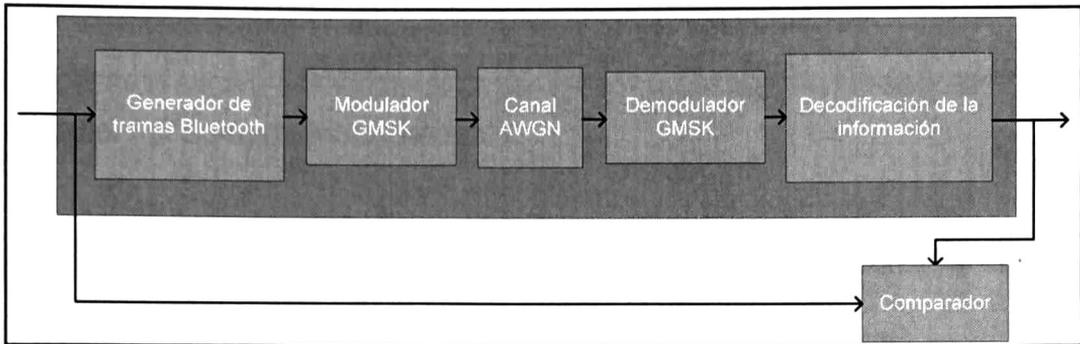


Figura 4.6 Diagrama de la interconexión de los bloques para las pruebas.

En este caso se definió el tipo de paquete en TYPE=5, que define un paquete HV1, este es un transporte lógico SCO el cual tiene 10 bytes de información útil. El índice de modulación del modulador GMSK es 0.32, señal generada por el modulador para por un canal AWGN con una relación señal a ruido de $E_b/N_0=12$ dB, el demodulador trabaja con un bloque de estimación de máxima semejanza considerando 2 bits en observación para tomar una decisión.

El paquete HV1 transporta 10 bytes de información, la palabra hexadecimal a transmitir es "799DC516F46CC51DF31F" con el "right-LSB" el generador de paquetes: forma el código de acceso, encabezado de paquete, realiza el whitening del encabezado de paquete y payload para evitar secuencias largas de 1's y 0's, posteriormente el payload lleva un codificación FEC de una tasa de 1/3.

Después de generar los paquetes de acuerdo a las especificaciones, los datos binarios se pasan por el modulador GMSK, teniendo las formas de ondas mostradas en la Figura 4.7, en el modulador se crean una rama en fase y otra en cuadratura para cada uno de los bits.

Una vez pasada la señal GMSK por el bloque que adiciona ruido Gaussiano, las formas de onda se modifican como se muestran en la Figura 4.8.

La trama transmitida vista en formato hexadecimal es:

07FFF1FF1C01F81F8007FF8E3FE3F1FF0071C0038E07FC71F8E07E3FE38FC
7FC0FC0FF8FC0AB1859432AD9632375.

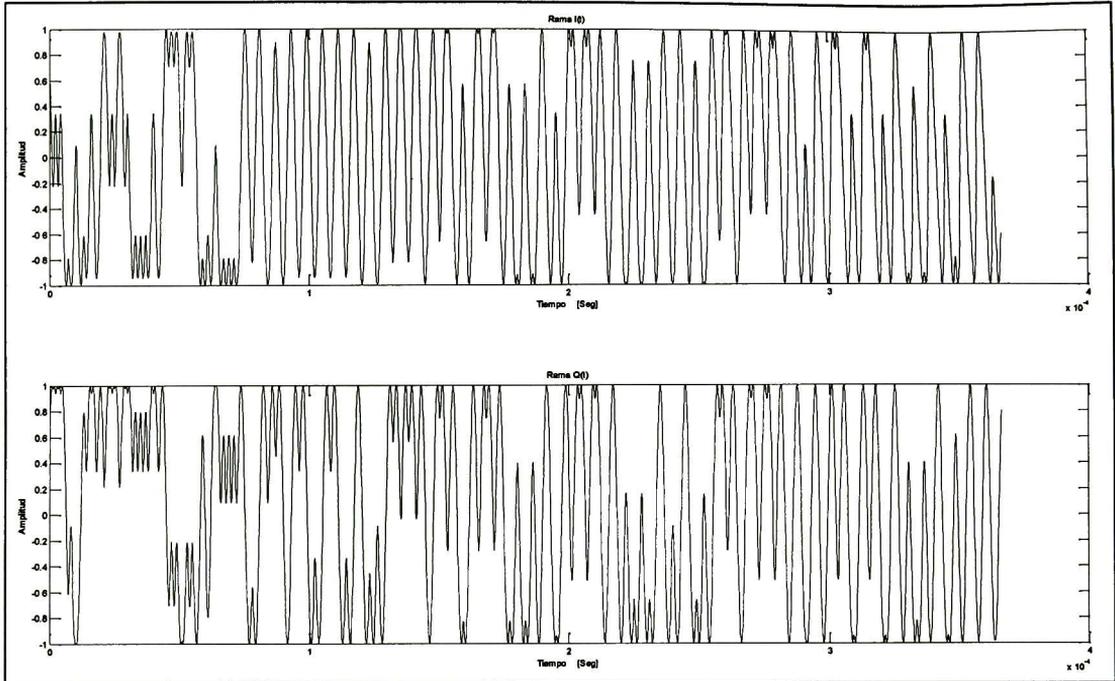


Figura 4.7 Formas de ondas de la señal GMSK en fase y en cuadratura para un paquete HV1.

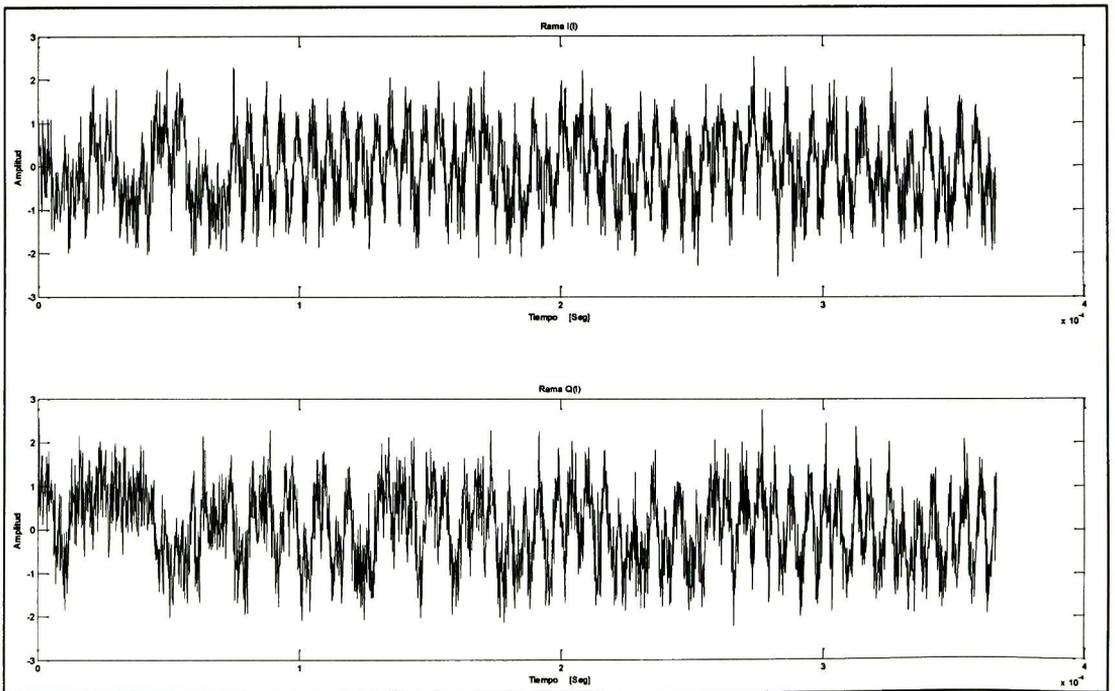


Figura 4.8 Formas de onda de la señal GMSK, para un paquete HV1 con ruido.

La trama recibida en formato hexadecimal con el "right-LSB" fue:

07FFF1FF1C01F81F8007FF8E3FE3F1FF0071C0038E07FC71F8E07E3FE38FC
7FC0FC0FF8FC0AB1859432AD9632375.

CAPÍTULO 5 CONCLUSIONES Y TRABAJO FUTURO

5.1 Conclusiones.

Para que el generador de tramas de Bluetooth se tuviera a punto se requirió de la lectura y comprensión del estándar Bluetooth V1.2, cabe hacer mención de que muchas de las partes del estándar se encuentran libres de interpretación, aunado al formato Little Endian utilizado, algunos valores proporcionados dentro del estándar generaron conflicto en su interpretación.

Muchas arquitecturas implementadas anteriormente para la demodulación de la señal GMSK, como el caso específico el demodulador GMSK basado en detector de cruce por ceros, resultó tener un desempeño bajo cuando la generación de las formas de onda GMSK se realizaban con moduladores implementados con tablas ROM.

Se realizó el comparativo entra varios moduladores y demoduladores para definir un par, que permitiera implementarse para el proyecto de ASTECAS.

De los moduladores encontrados en la bibliografía especializada se seleccionó el generador GMSK por tablas ROM, por su sencillez de implementación.

El demodulador propuesto es una extensión de las expresiones matemáticas propuestas en [24], para poder lograr la demodulación de la señal GMSK que no cubría este artículo, y definiendo la profundidad de la ventana de observación en $N=2$, que permite tener un balance entre el desempeño del demodulador GMSK y la complejidad de la arquitectura.

Se escogió el demodulador propuesto debido a que es una implementación robusta que permite recuperar la información al menos de los 3 moduladores propuestos, algo que no sucedió con el demodulador óptimo (Viterbi) dado que el desconocimiento exacto de los parámetros L y h del modulador no permiten la demodulación, esto se supero con el demodulador sub-óptimo propuesto

Se disminuyó la complejidad de la arquitectura del demodulador reduciendo las multiplicaciones por sumas y corrimientos.

Por medio del análisis en punto fijo se definió que son necesarios 10 bits de entrada para lograr la demodulación de la señal GMSK en banda base, y 13 bits de salida debido al efecto de los multiplicadores.

5.2 Trabajo futuro.

La implementación en hardware del procesamiento en banda base del estándar Bluetooth no estaba cubierto dentro de los objetivos de esta tesis, pero para el desarrollo del proyecto ASTECAS es imperativo su desarrollo.

De igual manera la implementación en hardware del modulador GMSK, demodulador GMSK, no se encuentran dentro de los objetivos de la tesis, por lo que se deja como trabajo futuro.

Dentro del proyecto ASTECAS se prevé la implementación de 3 estándares diferentes, un trabajo a desarrollar es ya teniendo los modelos de referencia de cada uno de los estándares, identificar los bloques comunes para ver la posibilidad de su reutilización.

La selección de alguno de los algoritmos de sincronía para usarse con el demodulador GMSK, así como la validación de su desempeño se plantea como un trabajo de futuro desarrollo, aunque existen trabajos previos en el grupo de telecomunicaciones sobre algoritmos de sincronización.

En este trabajo de tesis se desarrollaron 3 algoritmos de modulación de la señal GMSK y se desarrollaron 4 demoduladores; sería recomendable extender la búsqueda de algoritmos de generación y demodulación GMSK y compararlos contra los que se tienen implementados. En específico sería contrastar el desempeño del algoritmo de demodulación propuesto contra los demoduladores existentes que no se lograron comparar.

El desarrollo de la versión más reciente de Bluetooth V4.0, se puede plantear como un trabajo de tesis de maestría. Y si se considera el presente trabajo realizado, se deben de cambiar ciertas partes de la generación de tramas e incluir los otros dos formatos de modulación $\pi/4$ -DQPSK y el 8-DQPSK, así como el desarrollo del controlador de *Low Energy*.

GLOSARIO

Acknowledgement	Reconocimiento
Bias	Voltaje de polarización
Broadcast	Difusión
Butterfly	Mariposa
Host	Anfitrión
Inquiring	Indagación
Inquiry	Indagar
Kernel	Núcleo
Offset	Compensación
Page	Vocear en un canal de comunicaciones y entregar un mensaje.
Paged	Voceado
Pager	Dispositivo que hace el llamado sobre el canal y entrega un mensaje.
Paging	Llamado sobre un sistema de direccionamiento público para encontrar el destinatario y entregar un mensaje.
Park	parquear
Payload	Carga útil
Shaping	Formación
STANDBY	En espera
Trade-off	Concesiones mutuas
Throughput	Rendimiento
Trailer	Cola

ACRÓNIMOS

ACHS	Secuencia de saltos de canal adaptado.
ACK	Reconocimiento.
AFH	Salto de frecuencia adaptativo.
AR_ADDR	Dirección de solicitud de acceso.
BCHS	Secuencia de salto de canal básico.
BRM	Administrador de recursos de banda base.
CRC	Código de Redundancia Cíclica.
DC	Corriente continua.
DIAC	Código de Acceso de Inquiry Dedicado.
EDR	Tasa de datos mejorada (Por las siglas en inglés <i>Enhanced Data Rate</i>)
eSCO	Transporte lógico orientado a conexión síncrono extendido.
FHS	Sincronía de saltos de frecuencia.
FIFO	Registros en donde el primero en entrar es el primero en salir.
GIAC	Código de Acceso de Inquiry General.
GMSK	Gaussian Minimum Shift Keying
HEC	Código de Corrección de Encabezado.
IR	Inquiry scan – Escaneo de Inquiry
ISHS	Secuencia de saltos de escaneo de Inquiry
L2CAP	Protocolo de Adaptación y Control de Enlace Lógico
LAP	Parte de Dirección Baja
LC	Controlador de Enlace
LE	Baja Energía de la siglas en inglés Low Energy
LFSR	Registro de Corrimiento de Retroalimentación Lineal

ACRÓNIMOS

LLID	Identificador de Enlace Lógico
LSB	Bit Menos Significativo
LT_ADDR	Dirección de Transporte Lógico
MPRHS	Respuesta de page del maestro
Ms/s	Mega símbolos por segundo
MSB	Bit Más Significativo
NRZ	No retorno a cero
PM_ADDR	Dirección de miembro parqueado
PRNG	Generador de Ruido Pseudo-Aleatorio
PS	Escaneo de Page
PSD	Power Spectral Density
PSHS	Page scan - Escaneo de voceado
SCO	Transporte lógico orientado a conexión síncrono
SPRHS	Respuesta de page del esclavo
TS	Ranura de tiempo
UAP	Parte de Dirección Alta
VCO	Oscilador Controlado por Voltaje

BIBLIOGRAFÍA

- [1] Bluetooth SIG. (2011, Enero) Bluetooth. [Online].
https://www.bluetooth.org/About/bluetooth_sig.htm
- [2] (2011, Junio) HOOVERS A D&B COMPANY. [Online].
http://www.hoovers.com/business-information/--pageid_13751--/global-hoov-index.xhtml
- [3] BLUETOOTH SIG. (2011, ENERO) BLUETOOTH. [Online].
<http://www.bluetooth.com/Pages/History-of-Bluetooth.aspx>
- [4] SIG. (2003) Specification of the Bluetooth System V1.2.
- [5] CSR. (2011, Febrero) www.csr.com. [Online].
<http://www.csr.com/products/45/csr-energy>
- [6] Jeonghun Kim and Youngwhan Choi and Jungwon Jeong and Suhho Lee and Suki Kim, "The v2.0+EDR Bluetooth SOC architecture for multimedia," *IEEE_J_CE*, vol. 52, no. 436-444, 2006.
- [7] Seng Benny Tan Choon, Nios Softcore Base Design of the Bluetooth Lower, Octubre 2001.
- [8] Eynde Frank Op't, Jean-Jacques Schmit, and Vincent Charlier, "A Fully-Integrated Single-Chip SOC for Bluetooth," in *IEEE International Solid-State Circuits Conference*, 2001, p. 13.1.
- [9] B. A. and Nakaska, J. K. and Randall, R. G. and Haslett, J. W. Georgescu, "A 0.18 μ m CMOS Bluetooth frequency synthesizer for integration with a Bluetooth SOC reference platform," in *Proc. 3rd IEEE Int System-on-Chip for Real-Time Applications Workshop*, 2003.
- [10] A. and Hammes, M. Neubauer, "A Digital Receiver Architecture for Bluetooth in 0.25- μ m CMOS Technology and Beyond," *Circuits and Systems I: Regular*

- Papers, IEEE Transactions on*, vol. 54, no. 2044 -2053, 2007
- [11] K. Hirade, K. Murota, "GMSK Modulation for Digital Mobile Radio Telephony," vol. 29, no. 1044--1050, 1981.
- [12] J.G. Proakis, *Digital Communications*, 3rd ed. EUA: McGraw-Hill, 1995.
- [13] J. B. Anderson and T. Sundberg, C-E. Aulin, *Digital Phase Modulation*, 1st ed. N.Y.: Plenum Press, 1986.
- [14] Hermosillo G. J., *Estudio y simulación de algunos esquemas de modulación de fase continua (CPM)*, CINVESTAV, Ed. Guadalajara, 1999.
- [15] A. Y. Sanchez-Sinencio, E. Bo Xia and Chunyu Xin and Wenjun Sheng and Valero-Lopez, "A GFSK demodulator for low-IF Bluetooth receiver," *IEEE_J_JSSC*, vol. 38, pp. 1397--1400, 2003.
- [16] H. Khorram, S. Hung-Ming Chien and Meng-An Pan and Wu, S. and Moloudi, S. and Leete, J. C. and Rael, J. J. and Syed, M. and Lee, R. and Ibrahim, B. and Rofougaran, M. and Rofougaran, A. Darabi, "A 2.4-GHz CMOS transceiver for Bluetooth," *IEEE_J_JSSC*, vol. 36, no. 2016--2024, 2001.
- [17] Mototsugu Hamada, Ken-ichi Agawa, Shouhei Hiroki Ishikuro, "A Single-Chip CMOS Bluetooth Transceiver with 1.5 MHz IF and Direct Modulation Transmitter," , 2003.
- [18] Masaaki Shida, Takashi Ishikawa, Hiroki Sonoda, Katsumi Yamamoto, Tatsuji Matsuura, Masaharu Matsuoka, Takefumi Endo, Takao Kobayashi, Katsumi Oosaki, Takaaki Henmi, Junya Kudoh, Hirokazu Miyagawa Masaru Kokubo, "A 2.4GHz RF Transceiver with Digital Channel-Selection Filter for Bluetooth," *IEEE International Solid-State Circuits Conference*, 2002.
- [19] P. T. M. Eikenbroek, J.-W. Vervoort, P.-P. Setty, S. Tangenberg, J. Shipton, G. Kooistra, E. Keekstra, I. Belot, D. van Zeijl, "A Bluetooth radio in 0.18 μm CMOS," in *Proc. Digest of Technical Papers Solid-State Circuits Conf. ISSCC. 2002 IEEE Int*, 2002, pp. 1679 - 1687.

- [20] N. and Yalcin, T. Ismailoglu, "Low-power design of a digital FM demodulator based on zero-cross detection at IF," in *Fall Vehicular Technology Conference*, Amsterdam, 1999, pp. 810-813.
- [21] M. and Shida, M. and Ishikawa, T. and Sonoda, H. and Yamamoto, K. and Matsuura, T. and Matsuoka, M. and Endo, T. and Kobayashi, T. and Oosaki, K. and Henmi, T. and Kudoh, J. and Miyagawa, H. Kokubo, "A 2.4 GHz RF transceiver with digital channel-selection filter for Bluetooth," in *IEEE International Solid-State Circuits Conference*, 2002.
- [22] V. and Cho, T. and Kang, D. and Dow, S. Bang-Sup Song and Leung, "A 2.4 GHz Bluetooth transceiver in 0.18 μm CMOS," in *Proc. IEEE Asia-Pacific Conf. ASIC*, 2002, pp. 117-120.
- [23] Eric S. Otto and Phillip L. De León, "Digital CPFSK transmitter and noncoherent receiver/demodulator implementation," New Mexico State University, New Mexico, Article 1997.
- [24] M. K. and Divsalar, D. Simon, "Maximum-likelihood block detection of noncoherent continuous phase modulation," *IEEE TRANSACTIONS ON COMMUNICATIONS*, vol. 41, no. 1, pp. 90-98, January 1993.



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N.
UNIDAD GUADALAJARA**

"2011, Año del Turismo en México"

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Desarrollo del procesador de banda base del estándar Bluetooth

del (la) C.

Abisai RAMÍREZ PÉREZ

el día 15 de Diciembre de 2011.

Dr. Deni Librado Torres Román
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. Ramón Parra Michel
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Roberto Carrasco Álvarez
Profesor Investigador
Universidad de Guadalajara



CINVESTAV - IPN
Biblioteca Central



SSIT0010832